



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

MARCELO CABRAL BATISTA

**PROPOSTA DE UMA ONTOLOGIA APLICADA AO NIVEL G DO
MPS-SW**

**QUIXADÁ
2014**

MARCELO CABRAL BATISTA

**PROPOSTA DE UMA ONTOLOGIA APLICADA AO NIVEL G DO
MPS-SW**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação

Orientador Prof. Dr. Davi Romero de Vasconcelos

**QUIXADÁ
2014**

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca do Campus de Quixadá

B336p Batista, Marcelo Cabral
Proposta de uma ontologia aplicada ao nível G do MPS-SW / Marcelo Cabral Batista. – 2014.
46 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de
Sistemas de Informação, Quixadá, 2014.
Orientação: Prof. Dr. Davi Romero de Vasconcelos
Área de concentração: Computação

1. Software - Ontologia 2. Software de aplicação 3. Software – Desenvolvimento I. Título.

MARCELO CABRAL BATISTA

**PROPOSTA DE UMA ONTOLOGIA APLICADA AO NIVEL G DO
MPS-SW**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação

Aprovado em: _____ / dezembro / 2014.

BANCA EXAMINADORA

Prof. Dr. Davi Romero de Vasconcelos
Universidade Federal do Ceará-UFC

Prof. Msc. Camilo Camilo Almendra
Universidade Federal do Ceará-UFC

Prof. Msc. Enyo José Tavares Gonçalves
Universidade Federal do Ceará-UFC

A todos aqueles que contribuíram
na conquista deste objetivo

AGRADECIMENTOS

A toda a minha família, especialmente a minha mãe Lucia, meu pai João, meu irmão Marden, e minha amiga Ionara a quem considero da família, por sempre acreditarem em mim e darem-me o apoio necessário para seguir adiante.

Ao professor Davi Romero, por todos os ensinamentos que vão muito além da vida acadêmica. Pessoa e profissional íntegro que tive a felicidade conviver ao longo desses anos de graduação, pela paciência que teve com seu orientado, sempre respeitando minhas limitações e dando-me o tempo necessário para amadurecer ideias, pelo apoio e por todas as sugestões preciosas para esse trabalho.

A minha namorada Cristiane, por todo apoio, incentivo e carinho diante dos desafios encontrados.

Aos colegas de Graduação, em especial ao meu amigo Daniel Wolney, que me ajudou bastante ao longo da graduação.

Aos professores do campus, que ensinaram-me não só algoritmos, processos, mas também valores como respeito, compromisso, seriedade e criticidade.

Aos professores Ênyo José, Carla Ilane e Camilo Almendra por terem aceitado o convite para participar da banca de defesa deste trabalho de conclusão de curso.

Ao Grupo PET Sistemas de Informação onde fiz amizades que levarei por toda a vida e que me proporcionou a honra de conviver com colegas prudentes, ousados, divertidos, líderes, companheiros, proativos e guerreiros. Deram-me ensinamentos que me inspirarão por toda a vida.

"A imaginação muitas vezes conduz-nos a mundos a que nunca fomos, mas sem ela não iremos a nenhum lugar"

(Carl Sagan)

RESUMO

Criado em 2003, o programa Melhoria do Processo de Software Brasileiro (MPS.Br) propõe um modelo de processo para melhorar a capacidade de desenvolvimento de software nas empresas brasileiras. O MPS.Br está dividido em níveis de maturidade, representados por letras em ordem decrescente, com início de sua escala de maturidade no nível G, progredindo até o nível A. Neste trabalho é apresentada uma proposta de representação formal através de uma ontologia desenvolvida em lógica descritiva que aborda os conceitos relevantes às áreas de processos contidas no nível G do guia Melhoria de Processo do Software Brasileiro (MPS.Br-SW). A modelagem da ontologia foi realizada mediante a utilização da linguagem Ontology Web Language (OWL) e do framework Protégé 4.3, pelo fato do mesmo dar suporte à linguagem utilizada para construção de Ontologias e fazer uso de máquinas de raciocínio onde podemos verificar a consistência e correteza da Ontologia construída. Através da representação em ontologia, proposta por este trabalho, busca-se facilitar a compreensão dos processos contidos no nível G do MPS-SW e tê-la como um mecanismo de apoio para empresas desenvolvedoras de software que buscam certificação nesse nível.

Palavras-Chave: Ontologia. MPS.Br. Nível G.

ABSTRACT

Created in 2003, the Brazilian Software Process Improvement program (MPS.Br) proposes a process model to Improve software development capacity in the Brazilian companies. The MPS.Br is divided in maturity levels, Represented by letters in descending order, with begin of its scale of maturity in the level G, progressing until level A. This work presents a proposal of formal representation through an ontology developed in description logics that shows the relevant concepts the process areas contained in level G of the Brazilian Software Process Improvement Program (MPS.Br-SW). The modeling of the ontology was carried by using the Ontology Web Language (OWL) and the framework Protégé 4.3 , by it give support to the language used for building ontologies and make use of reasoning machines where is possible check the consistency and regularity of the Ontology built. Through representation in ontology, the present work seek to facilitate the understanding of the processes contained in the G level of the MPS-SW and has it as a support mechanism to software development companies seeking certification at this level.

Keywords: Ontology. MPS Br. Level G.

LISTA DE TABELAS

Tabela 1 - Conjunto de guias com orientações para implementação do MPS-SW	20
Tabela 2 - Processos e Atributos de Processo do Nível G.....	21
Tabela 3 - Descrição dos Atributos de Processo Nível G 1.1 e 2.1	22
Tabela 4 - Resultados esperados do processo de Gerência de Projetos do MPS.Br nível G	23
Tabela 5 - Resultados esperados do processo de Gerência de Requisitos do MPS.Br nível G	24
Tabela 6 - Descrição das principais classes da Ontologia proposta	38

LISTA DE ILUSTRAÇÕES

Figura 1: Componentes do MPS.BR.....	19
Figura 2: Níveis de maturidade do MPS.BR	20
Figura 3: Tipos de Ontologia	27
Figura 4: Interface principal da ferramenta Protégé	30
Figura 5: Processo iterativo de construção da ontologia	33
Figura 6: Superclasses e Subclasses da Ontologia Proposta	38
Figura 7: Processos Contidos na Gerência de Projetos	39
Figura 8: Propriedades criadas para relacionar as classes e suas respectivas propriedades	40
Figura 9: Inter-relacionamento entre duas classes, proporcionado pela propriedade inversa ..	40
Figura 10: Propriedade inferindo classe	41
Figura 11: Representação de Propriedade inferindo classe	41
Figura 12: Descrição que explica o propósito da Classe	42
Figura 13: Texto Atributos de Processo no Nível G.....	42
Figura 14: Descrição dos Atributos	43
Figura 15: Hierarquia das Classes do Domínio Capacidade do Processo	44
Figura 16: Criação de Classe Inconsistente	45
Figura 17: A subclasse Atividades destacada	46
Figura 18: Visão parcial das classes construídas automaticamente pelo Reasoner sem apontamento de inconsistência	47
Figura 19: Consulta utilizada para verificar qual processo tem como resultado o Ciclo de Vida do Projeto.....	47
Figura 20: Consulta utilizada para verificar qual processo tem como resultado Revisões dos Requisitos	48
Figura 21: Processo gerência de Projeto 01 (GPR1)	49
Figura 22: Modelagem Ontologia Proposta.....	50
Figura 23: Hierarquia das Classes	51
Figura 24: Hierarquia do GPR1 Pizzoleto (2013)	52
Figura 25: Definição do Nível G do Pizzoleto (2013).....	52
Figura 26: Definição do Nível G - Ontologia Proposta.....	54
Figura 27: Características do Nível G	55
Figura 28: Comparação da Classe Gerência de Projetos	56
Figura 29: Comparando a classe Gerência de Requisitos.....	57
Figura 30: Modelagem da Classe Desafios	57
Figura 31: Comparação das Classes Principais das Ontologia	58

SUMÁRIO

1 INTRODUÇÃO	15
2 FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 MPS.BR	18
2.1.1 O Nível G – (Parcialmente Gerenciado).....	21
2.2 Ontologia de domínio para representação do conhecimento	25
2.2.1 Elementos de uma ontologia.....	26
2.2.2 Tipos de Ontologia	27
2.2.3 Linguagens usadas para representação de Ontologias	28
2.2.4 Ferramenta para construção da Ontologia	29
3 TRABALHOS CORRELATOS	30
3.1 DESENVOLVIMENTO DA ONTOLOGIA.....	32
4 PROCEDIMENTOS METODOLÓGICOS	33
5 CONSTRUÇÃO DA ONTOLOGIA	36
5.1 CRIAÇÃO DAS CLASSES	37
5.2 PROPRIEDADES DE OBJETO.....	39
5.3 DESCRIÇÃO DAS CLASSES NA FERRAMENTA PROTÉGÉ.....	42
5.4 VERIFICAÇÃO DA ONTOLOGIA	44
5.4.1 DEMONSTRAÇÃO DE DETECÇÃO DE INCONSISTÊNCIA	45
5.4.2 VERIFICAÇÃO DE INCONSISTÊNCIA NA ONTOLOGIA.....	46
5.4.3 VERIFICANDO EQUIVALÊNCIAS ATRAVÉS DE CONSULTAS	47
5.5 DIFERENÇAS ENTRE OS TRABALHOS	48
5.6 COMPARANDO AS MODELAGENS	55
6 CONSIDERAÇÕES FINAIS	59
7 REFERÊNCIAS	61

1 INTRODUÇÃO

Um modelo de qualidade fornece uma estrutura de comportamento para as organizações gerarem produtos que satisfaçam as necessidades dos seus clientes. O modelo, abrange não somente os produtos, como também os serviços e, especialmente, os processos que geram produtos e serviços (BRANDÃO LIMA, 2004). A maioria dos modelos de qualidade de software são hierárquicos, geralmente definidos em níveis de maturidade que estabelecem patamares de evolução e estágios de melhorias de processos, isto é, definem características que agrupam subcaracterísticas. Essas subcaracterísticas definem atributos que são utilizados para avaliar o processo de desenvolvimento do software.

O programa Melhoria do Processo de Software Brasileiro (MPS.Br) foi criado em 2003, com o objetivo de propor um modelo de processo para melhorar a capacidade de desenvolvimento de software nas empresas brasileiras (KOSCIANKI, 2007). Ele é escrito em língua portuguesa, com vocabulário destinado à micro e pequenas empresas desenvolvedoras de software de diferentes categorias funcionais e tamanhos, bem como para diferentes perfis de pessoas envolvidas e está dividido em quatro componentes: Modelo de Referência MPS para Software - MR-MPS-SW; Modelo de Referência MPS para Serviços - MR-MPS-SV; Método de Avaliação - MA-MPS e Modelo de Negócio MN-MPS.

O MPS.Br possui sete níveis de maturidade que são representados por letras em ordem decrescente com início de sua escala de maturidade no nível G, progredindo até o nível A. Cada nível possui um perfil de processos, com o objetivo de indicar onde uma organização deve concentrar esforços na busca por melhorias e evoluções. De acordo com o Guia de Implementação MPS.Br (2009i), o nível G, tem por objetivo nortear a organização para que ela seja capaz de gerenciar parcialmente seus projetos de desenvolvimento de software. Ele é considerado o nível que gera maior dificuldade em sua implantação devido às dificuldades que as empresas possuem em entender o significado de processo e também pela mudança cultural que ocorre na empresa quando se opta por seguir um padrão de desenvolvimento (SOFTEX 2007a).

Implementar processos em uma organização não é uma tarefa trivial. Cada um dos processos geralmente é complexo, “devido à grande quantidade de atividades a serem executadas e informações envolvidas” (BERTOLLO; FALBO, 2002). De acordo com Pizzoleto (2013), há grande dificuldade de uniformizar o entendimento por parte das pessoas envolvidas na implementação, consultoria e certificação desses modelos, devido à grande

diversidade, quantidade de conteúdo e interdependência, por conta do vasto conjunto de informações, tanto em abrangência como em profundidade contido neles.

Uma alternativa para organização desse conjunto de informações é através do uso de ontologias. Segundo Guarino (1998), ontologia vêm sendo amplamente utilizada como técnica de representação e reutilização do conhecimento e de forma simplificada, pode-se dizer que é um conjunto de conceitos e termos usados para descrever determinado domínio, que se relacionam por atributos e relacionamentos. Ontologias consistem em uma alternativa para caracterizar e relacionar entidades em um domínio, representando o conhecimento nele contido, tornando-se um importante meio de representar, formalizar e compartilhar conhecimento. Através do uso de ontologias, é possível modelar um domínio de conhecimento descrevendo seus conceitos e relacionamentos. O formalismo adotado por elas, associado a um domínio específico, permite representar o conhecimento sem ambiguidades e facilita a interpretação sobre um dado conceito relacionado ao domínio que está sendo mapeado, assegurando qualidade às inferências realizadas. Essas qualidade, podem ser verificada através de mecanismo de raciocínio (reasoner) que são importantes para testar a consistência da ontologia, checando a taxonomia das classes e a consistência das propriedades e restrições, além de verificar se as declarações e as definições da ontologia são mutuamente consistentes entre si e reconhecer se os conceitos são adequados as definições.

Nesse sentido, uma ontologia, fornece uma descrição exata do conhecimento. Diferentemente da linguagem natural em que as palavras podem ter semântica totalmente diferente conforme o seu contexto, a ontologia por ser escrita em linguagem formal, não deixa espaço para o gap semântico existente na linguagem natural. Por exemplo, quando uma pessoa fala para outra a palavra “Globo” ela pode estar querendo falar a respeito de um corpo esférico, como também de um canal de televisão brasileiro. A interpretação da palavra pode ser atribuída a um conceito ou outro conforme o estado mental do indivíduo. Porém, se há uma conceitualização comum entre essas duas pessoas a possibilidade de mal entendido diminui muito. Por exemplo, se essas pessoas concordam em uma ontologia sobre o domínio de formas geométricas, possivelmente não haverá mal entendido.

Quando uma ontologia faz uso de uma linguagem lógica, esta passa a ser reconhecida como uma ontologia formal. Nesse trabalho, o principal objetivo consiste em elaborar uma representação formal através de uma ontologia em lógica descritiva, que aborda os conceitos relevantes às áreas de processos contidas no nível G, para organizar, simplificar, apoiar e facilitar a compreensão dos processos envolvidos nesse nível do guia Melhoria de Processo do Software Brasileiro (MPS-SW) e garantindo as distinções entre as classes, propriedades e

relações da ontologia, evitando ambiguidades. Através da representação em ontologia, busca-se facilitar a compreensão dos processos contidos nesses níveis e tê-la como um mecanismo de apoio para empresas desenvolvedoras de software que buscam certificação nesse nível. Busca-se também, formalizar o procedimento de implantação do nível G através da definição de uma linguagem formal em lógica descritiva; propõe-se uma forma alternativa de organizar o conhecimento do MPS.Br, nível G, através da definição de uma ontologia e permitir que futuros usuários formulem consultas usando os conceitos especificados de acordo com as especificações da linguagem definida.

Para este trabalho, foram realizadas pesquisas e constatou-se que poucos trabalhos tem tratado de ontologias direcionadas ao domínio MPS.Br. e um número considerável de trabalhos que abordam o domínio de Engenharia de Software. O trabalho de Bertollo e Falbo (2005) propõem uma ontologia de Processos de Software, que foi graficamente representada utilizando UML e tem como objetivo fornecer a compreensão, por parte das organizações de desenvolvimento de software, dos conceitos relacionados com processos de software incluindo padrões e metodologias. Souza (2012) aborda o uso de Ontologias na Garantia da Qualidade e Melhoria de Processos, onde é apresentada uma visão geral do estudo da arte das ontologias e seu uso na garantia da qualidade e na melhoria de processos de software. O trabalho de Cunha (2011) não está relacionado a construção de ontologias. Através de um estudo de caso, Cunha (2011) apresenta as Dificuldades Encontradas na Implementação do nível G do MPS.BR e explora as dificuldades encontradas durante a implementação desse nível em algumas empresas, onde são registradas as dificuldades, vantagens e lições aprendidas durante sua implementação.

O trabalho que mais se aproxima do trabalho proposto é o de Pizzoleto (2013), que apresenta uma ontologia empresarial aplicada aos níveis G e F do MPS-SW. Para o desenvolvimento de sua ontologia, Pizzoleto (2013) faz uma análise das estruturas dos treze guias do modelo MPS-SW, com o propósito de encontrar uma estrutura comum entre eles. Desenvolve também, uma metodologia baseada em ontologia empresarial que define o processo de mapeamento das informações contidas nesses guias e através de um levantamento, insere conceitos e terminologias contemplados no Guia de Conhecimentos em Gerenciamento de Projetos ou PMBOK (Project Management Body of Knowledge), e também considera a inclusão de indicadores do modelo BSC (Balanced Scorecard), um modelo de mensuração de desempenho empresarial para complementar o grupo de indicadores do modelo MPS-SW.

A seleção desse nível foi motivada por se tratar no nível inicial para uma empresa que deseja obter uma certificação MPS.Br, pelas dificuldades encontradas em compreender o significado dos processos no nível G, por conta da quantidade de conteúdo contida nesse nível e pelos obstáculos na sua implantação em uma empresa que inicia um processo de maturidade utilizando o MPS.Br como modelo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados alguns conceitos importantes que foram utilizados e que facilitam a compreensão do trabalho apresentado. A seção 2.1 apresenta o modelo de maturidade MPS.Br, como ele está estruturado e a forma hierárquica como estão organizados seus níveis de maturidade. A subseção 2.1.1 descreve o nível G do MPS-SW, os processos e atributos contidos nesse nível. A Seção 2.2 descreve o conceito de ontologia aplicada a ciências da computação, sua importância na representação do conhecimento e seus benefícios na reutilização do conhecimento. A subseções 2.2.1 fala dos elementos que fazem parte de uma ontologia, a 2.2.2 dos tipos de ontologia existentes, 2.2.3 das linguagens utilizadas para construção de uma ontologia e a 2.2.4 das ferramentas utilizadas para construção de uma ontologia.

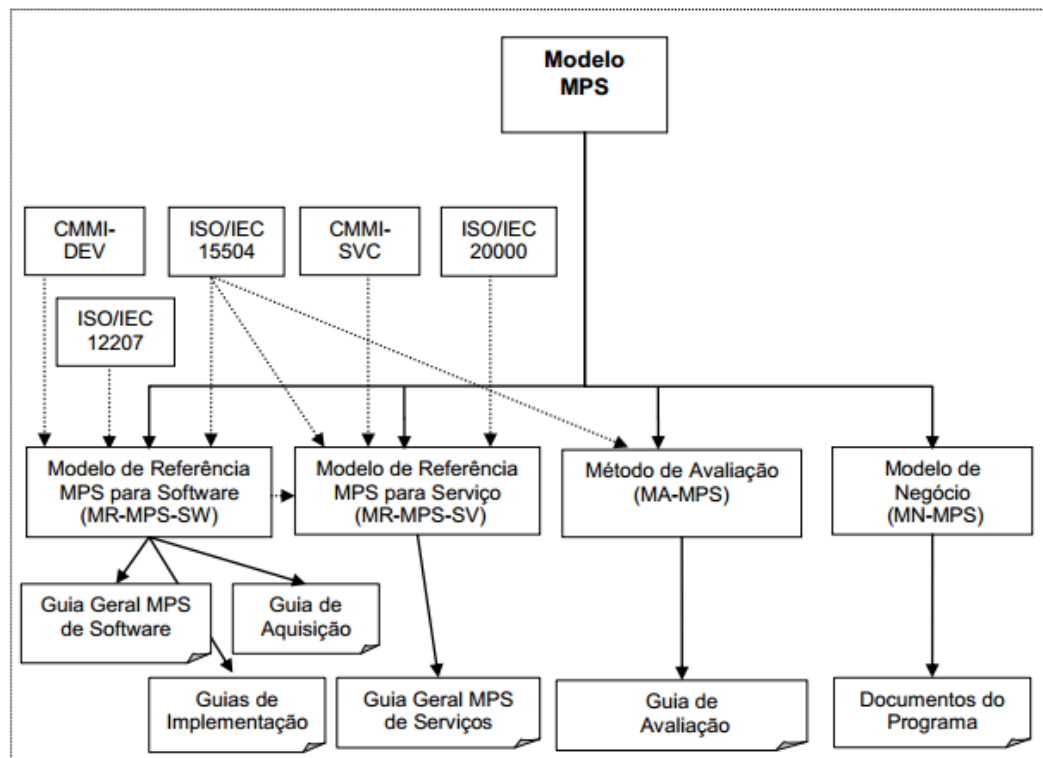
2.1 MPS.BR

O MPS.BR é um programa para Melhoria de Processo do Software Brasileiro que foi criado em 2003 e é coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), contando com apoio do Ministério da Ciência, Tecnologia e Inovação (MCTI), da Financiadora de Estudos e Projetos (FINEP), Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e do Banco Interamericano de Desenvolvimento (BID) (SOFTEX, 2012a). Uma das metas do Programa MPS.Br é definir e aprimorar um modelo de melhoria e avaliação de processo de software e serviços, visando preferencialmente às micro, pequenas e médias empresas (mPME) (SOFTEX, 2012a).

No contexto de seu modelo, o MPS.Br possui em sua base, o modelo CMMI (Capability Maturity Model Integration) e as normas ISO/IEC 12207:2008 (Processo de Ciclo de Vida de Software), ISO/IEC 15504-2 (Avaliação de Processo) e a ISO/IEC 20000 (Gerenciamento de Serviço). Atualmente no guia (SOFTEX, 2012a), o MPS.Br está dividido

em quatro componentes: MR-MPS-SW, contém as definições dos níveis de maturidade, processos e atributos do processo; MR-MPS-SV com as definições dos níveis de maturidade, processos e atributos do processo; MA-MPS, descreve um conjunto de atividades e tarefas para verificar a maturidade da unidade organizacional na execução dos seus processos de software e o MN-MPS descreve as regras de negócio para implementação do MR-MPS-SW e MR-MPS-SV, pelas Instituições Implementadoras. A Figura 1 mostra a estrutura dos componentes do MPS.Br.

Figura 1. Componentes do MPS.BR



FONTE: (SOFTEX, 2012a)

O MPS baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade. Os níveis de maturidades são representados por letras do alfabeto, de A a G, em ordem decrescente, com início de sua escala de maturidade no nível G, progredindo até o nível A, conforme Figura 2, que exemplifica a estrutura hierárquica dos níveis de maturidade no MPS.Br.

Figura 2: Níveis de maturidade do MPS.BR



Fonte: FUMSOFT (2012).

O MPS.Br está dividido em sete níveis de maturidade, definidos como: G (Parcialmente Gerenciado), F (Gerenciado), E (Parcialmente Definido), D (Largamente Definido), C (Definido), B (Gerenciado Quantitativamente) e A (Em Otimização). Eles denotam a evolução da empresa, estabelecem patamares de evolução e definem onde as empresas devem concentrar os esforços para implementação de melhorias. Para cada nível é necessário cumprir vários atributos e processos.

O Modelo de Referência MPS-SW, contém orientações para sua implementação nas empresas desenvolvedoras de software. Ele conta com um conjunto de treze (13) guias de caráter específico, definidos em sete níveis de maturidade para sua implementação, com início de sua escala no nível G, até o nível A, conforme Tabela 1 abaixo.

Tabela 1 - Conjunto de guias com orientações para implementação do MPS-SW

IDENTIFICAÇÃO DO GUIA	PROPÓSITO	ANO
Parte 1	Nível G	Setembro 2013
Parte 2	Nível F	Setembro 2013
Parte 3	Nível E	Setembro 2013
Parte 4	Nível D	Setembro 2013
Parte 5	Nível C	Setembro 2013
Parte 6	Nível B	Setembro 2013
Parte 7	Nível A	Setembro 2013
Parte 8	Organizações que adquirem software (Níveis G a A)	Novembro 2011
Parte 9	Organizações do tipo Fábrica de Software (Níveis G a A)	Novembro 2011
Parte 10	Organizações do tipo Fábrica de Teste (Níveis G a A)	Novembro 2011
Parte 11	Para a implementação e avaliação do Modelo de Referência MR-MPS-SW:2012 em conjunto com o CMMI-DEV v1.3.	Agosto 2012
Parte 12	Análise de aderência do Modelo de Referência MR-MPS-SW:2012 em relação à NBR ISO/IEC 29110-4-	Setembro 2012

1:2012.

Parte 13	Mapeamento e sistemas de equivalências entre o Modelo de Referência MR-MPS-SW e o MoProSoft.	Outubro 2012
----------	----------------------------------------------------------------------------------------------	--------------

FONTE: Adaptado pelo autor

De acordo com o Guia Geral do MR-MPS (SOFTEX, 2009a) os níveis de maturidade são uma combinação entre processos e sua capacidade. A definição de processo segue os requisitos da norma ISO/IEC 15504-2, declarando o propósito e os resultados esperados do processo. A capacidade é caracterizada pela habilidade do processo em alcançar os objetivos do negócio, relacionado ao atendimento dos resultados dos atributos de processo por nível de capacidade (SOFTEX, 2012a). O modelo define dezenove processos, agrupados nos sete níveis de maturidade. Nesses níveis, há um grande número de dependências entre as informações em um mesmo nível e entre os níveis de maturidade.

2.1.1 O Nível G – (Parcialmente Gerenciado)

Esse nível corresponde à primeira parte do processo de melhoria na organização, sendo formado pelos processos de Gerência de projetos (GPR) e Gerência de Requisitos (GRE). Ambos contém tópicos que são chamados de resultados esperados e que devem ser praticados pela empresa ao longo do processo de desenvolvimento de software. Cada nível do MPS.Br, possui uma distribuição de atributos de processos. No nível G, há dois Atributos de Processos (AP), O AP 1.1 mede se o processo é executado e AP 2.1 se o processo executado é gerenciado. Na Tabela 2 podemos ver a divisão dos processos e atributos de processo do nível G.

Tabela 2 - Processos e Atributos de Processo do Nível G

Nível	Processos	Atributos de processo
G	Gerência de Requisitos – GRE	AP 1.1 e AP 2.1
	Gerência de Projetos – GPR	

Fonte: elaboração do autor.

De acordo com (MPS.BR 2006a), a capacidade do processo é definida pelos atributos do processo em termos dos resultados esperados. Para se obter um nível de maturidade, a unidade organizacional deve atender aos Atributos do Processo (AP). Esses são definidos pelos resultados esperados dos atributos do processo (RAP). A Tabela 3 mostra os atributos do processo previstos para o nível G seguido de seus resultados esperados do atributo do processo, que também estão relacionados com o nível G. A primeira coluna mostra o atributo

do processo com seu nome e identificador. A segunda mostra os identificadores dos resultados esperados para aquele atributo. A terceira coluna mostra a descrição do RAP associada ao identificador de sua esquerda.

Tabela 3 - Descrição dos Atributos de Processo Nível G 1.1 e 2.1

ATRIBUTO DO PROCESSO	IDENTIFICADOR DO RAP	DESCRIÇÃO DO RAP	
AP 1.1 – O processo é executado	RAP1	O processo atinge seus resultados definidos	
	RAP2	Existe uma política organizacional estabelecida e mantida para o processo	
	RAP3	A execução do processo é planejada	
	RAP4	A execução do processo é monitorada e ajustes são realizados para atender o plano	
	RAP5	O recursos necessários para a execução do processo são identificados e disponibilizados.	
	RAP6	As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência	
	AP 2.1 – O processo é gerenciado	RAP7	A comunicação entre as partes interessadas no processo é gerenciada de forma a garantir seu envolvimento no projeto
		RAP8	O estado, atividades e resultados do processo são revistos com os níveis adequados de gerência (incluindo a gerência de alto nível) e problemas pertinentes são tratados

Fonte: Adaptado de (SOFTEX, 2011).

O Nível G é descrito em termos de propósito e resultados. O propósito descreve o objetivo geral a ser atingido durante a execução do processo e os resultados esperados estabelecem o que será obtido com a efetiva aplicação do processo.

Nesse nível de maturidade, a área de gerência de projetos (GPR) tem como objetivo principal, estabelecer e manter planos, definir atividades, recursos e responsabilidades do projeto e corrigir desvios no desempenho do mesmo, com propósito de fazer com que a organização seja capaz de planejar atividades, recursos e responsabilidade do projeto assim como gerenciar informações sobre o andamento do projeto, possibilitando intervenção dos responsáveis quando necessário. À medida que a organização avança nos níveis de maturidade, o processo incorpora novos resultados esperados. Dentro do GPR, estão contidas atividades como desenvolver um plano de projeto, obter e manter o comprometimento dos envolvidos com o projeto e acompanhar o andamento do projeto tomando ações corretivas quando necessário. De acordo com o guia de implementação (SOFTEX, 2012), este processo possui 19 (dezenove) resultados esperados para o nível G do MR.MPS conforme Tabela 4, onde são listados os 19 (dezenove) resultados esperados do processo de Gerência de Projetos e seus respectivos objetivos.

Tabela 4 - Resultados esperados do processo de Gerência de Projetos do MPS.Br nível G

RESULTADO	PROPÓSITO
GPR01	O escopo do trabalho para o projeto é definido.
GPR02	As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados.
GPR03	O modelo e as fases do ciclo de vida do projeto são definidos.
GPR04	O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas.
GPR05	O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos.
GPR06	Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados.
GPR07	Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo.
GPR08	Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados.
GPR09	Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança.
GPR10	Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos.
GPR11	A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados.
GPR12	O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido e mantido.
GPR13	O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado.
GPR14	Os recursos materiais e humanos, bem como os dados relevantes do projeto são monitorados em relação ao planejado.
GPR15	Os riscos são monitorados em relação ao planejado.
GPR16	O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido.
GPR17	Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento.
GPR18	Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas.
GPR19	Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.

Fonte: Adaptado de (SOFTEX, 2011).

Na Gerência de Requisitos (GRE), de acordo com (SOFTEX, 2011), o principal objetivo é controlar a evolução dos requisitos e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto. O processo Gerência de Requisitos (GRE) gerencia todos os requisitos recebidos ou gerados pelo projeto, incluindo requisitos funcionais e não-funcionais, bem como os requisitos impostos ao projeto pela organização. A organização deve assegurar que os requisitos acordados são gerenciados, suportando o planejamento e execução do projeto, através de passos apropriados. Quando os requisitos são recebidos através do fornecedor de requisitos, a organização deve garantir o entendimento comum entre as partes. Isso deve ser feito através de revisões, até que se chegue a um acordo, onde se obtêm um compromisso tanto do fornecedor de requisitos quanto da equipe de projeto. Este processo possui 05 (cinco) resultados esperados para o nível G do MR.MPS conforme Quadro 4.

Tabela 5 - Resultados esperados do processo de Gerência de Requisitos do MPS.Br nível G

RESULTADO	PROPÓSITO
GRE01	Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando critérios objetivos
GRE 02	O comprometimento da equipe técnica com os requisitos aprovados é obtido;
GRE 03	A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
GRE 04	Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
GRE 05	Mudanças nos requisitos são gerenciadas ao longo do projeto.

Fonte: Adaptado de (SOFTEX, 2011).

Com todos esses processos e atributos, segundo (SOFTEX, 2007a), esse nível é considerado o de maior dificuldade em sua implantação, devido às dificuldades que as empresas têm em entender o significado de processo e também pela mudança cultural que ocorre na empresa quando se opta por seguir um modelo de desenvolvimento. O objetivo da implantação deste nível é nortear a organização para que ela seja capaz de gerenciar parcialmente seus projetos de desenvolvimento de software.

Apesar de ser o nível mais baixo de referência no guia, o nível de maturidade G requer cuidados especiais na sua implementação. O início do trabalho para implantação de melhorias no software inclui desafios na definição de “projeto” e requer que os processos correspondentes a esse nível, sejam compreendidos de forma clara e coerente. Nesse nível, o papel fundamental para que a organização siga os processos de forma correta é do gerente de

projetos. Assim, uma má interpretação no guia de processos correspondentes a esse nível, pode ocasionar custos e tempo para organização.

2.2 Ontologia de domínio para representação do conhecimento

A palavra ontologia originalmente foi proposta por filósofos e deriva do grego *ontos* (ser) + *logos* (palavra) e significa o estudo do ser, dedicando-a a natureza e à existência de elementos. De acordo com (KIRYAKOV, 2006), é um termo que tem diferentes definições nas disciplinas da Filosofia e de Ciência da Computação.

Para Ciência da computação possui um outro significado, e em sua primeira definição mais popular, definida por (GRUBER, 1993), é definida como uma especificação explícita de uma conceitualização, uma definição genérica que à considera apenas como uma descrição de conceitos e definições. Borst (1997) faz uma definição mais específica, definindo-a como uma especificação formal, algo que é legível para computadores, com conceitos, relações, funções, propriedades, restrições e axiomas explicitamente definidos, ou seja, uma descrição de conceitos, definições, propriedades, restrições e relações, descritas na forma de axiomas legíveis por computadores. De forma simplificada, pode-se dizer que uma ontologia é um conjunto de conceitos e termos usados para descrever determinado domínio que se relacionam por atributos e relacionamentos (GUARINO, 1998). Segundo Almeida e Bax (2003), a organização de uma ontologia está baseada em uma estrutura de conceitos e seus relacionamentos com os seguintes componentes básicos:

- Classes: conceitos da ontologia organizados hierarquicamente. As classes superiores podem ser chamadas de superclasses e as classes inferiores de subclasses;
- Relações: interação entre os conceitos de um determinado domínio;
- Axiomas: especificações de restrições entre os conceitos e suas relações;
- Instâncias: representações de elementos específicos.

Ontologias vêm sendo amplamente utilizadas como técnica de representação e reutilização do conhecimento ao longo dos anos. Uma razão para sua popularidade existe devido à grande promessa de compartilhamento e entendimento comum de algum domínio de conhecimento que pode ser comunicado entre pessoas e computadores. Neste sentido, ontologias têm sido desenvolvidas para facilitar o compartilhamento e reutilização de informações (GRUBER, 2013).

Atualmente, muitos trabalhos na área de tecnologia vêm explorando seu uso em diversas áreas como: Processamento de Linguagem Natural, como Oncoterm, por exemplo,

que facilita a tradução de textos médicos sobre oncologia mediante uma ontologia baseada em textos especializados e dicionários médicos; Educação, com o SchoolOnto (Scholarly Ontologies Project) que é um biblioteca digital baseada em ontologias, dentre outros (PIZZOLETO, 2013).

2.2.1 Elementos de uma ontologia

Para Falbo e Duarte (2000), uma ontologia consiste basicamente dos conceitos e relações que existem em um determinado domínio, suas definições e propriedades, além de restrições descritas na forma de axiomas.

Chandrasekaran, Josephson e Benjamins (1999) afirmam que uma ontologia deve possuir os seguintes itens:

- Objetos, que são os termos que a ontologia deve proporcionar;
- Propriedades ou Atributos, que assumem valores e ajudam a explicar o conceito do objeto;
- Relacionamentos entre os vários objetos;
- Eventos, que podem ocorrer em determinados instantes em um dado objeto;
- Processos, dos quais os objetos da ontologia participam;
- Estados em que os objetos podem se encontrar;
- Efeitos causados pelos eventos, os quais podem ser outros eventos ou gerar novos estados.

Ainda segundo os autores, a representação dos objetos, seus relacionamentos, estados eventos e processos deve estar necessariamente inserida em um determinado domínio, sem o qual esses itens não têm sentido. Por exemplo, modelar os objetos Animal, Mineral e Planta não teria sentido algum sem um domínio específico. Porém, ao colocar tais objetos no domínio Planeta Terra, tais objetos passam a ter sentido como sendo os seus elementos constituintes. Atributos podem ser definidos para os conceitos a fim de caracterizá-los. Por exemplo, podem ser definidos os atributos Possui_Vida(x), onde X pode assumir animal e planta ou Contem_Carbono(x), informando se Animal, Mineral ou Planta é composto por carbono. Como exemplo de relacionamento, tem-se Pode_Comer(x,y), informando que o elemento x (por ex., Animal) pode comer o elemento y (por ex., Planta) (CHANDRASEKARAN; JOSEPHSON; BENJAMINS, 1999). Os objetos, propriedades e

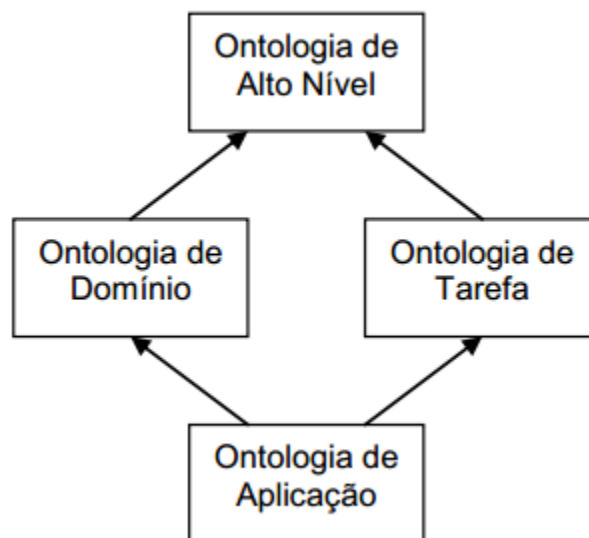
relacionamentos são específicos desse domínio e devem necessariamente refletir essa realidade.

2.2.2 Tipos de Ontologia

Há várias propostas de classificação de Ontologias. De acordo com Guarino (1998) classifica as ontologias segundo seu nível de generalidades: ontologias de alto nível, de domínio, de tarefas e de aplicação. Em sua classificação, as ontologias de alto nível, podem ser consideradas genéricas, sendo aplicáveis a qualquer domínio, podendo ser domínios particulares como Tecnologia da Informação, Medicina e automobilismo por exemplo, ou ontologias aplicáveis a tarefas genéricas, sem um domínio particular. As Ontologias de tarefas descrevem vocabulários relacionados a tarefas ou atividades genéricas, através da especialização de conceitos introduzidos nas ontologias de alto nível. As ontologias de domínio descrevem o vocabulário relacionado a um domínio específico, como medicina ou automobilístico, por exemplo. Por fim, as ontologias de aplicação são as mais específicas, usadas dentro de aplicações e podem especializar tanto conceitos das ontologias de tarefa quanto das ontologias de domínio (GUARINO, 1998).

Na Figura 3, tal como ilustra, pode-se observar visualmente que as ontologias de aplicação são especificações das ontologias de domínio e/ou das ontologias de tarefas que, por sua vez, são especificações das ontologias genéricas ou de alto nível.

Figura 3: Tipos de Ontologia



Fonte: GUARINO (1998)

2.2.3 Linguagens usadas para representação de Ontologias

Um aspecto que deve ser levado em consideração na construção de ontologias são as linguagens de representação. Segundo Oliveira (1999), as linguagens são “um ponto importante para operacionalizar a ontologia, ou seja, permitir sua efetiva utilização”. As linguagens mais recentes de representação de ontologias são baseadas em Lógica descritiva, das quais se destacam:

- OIL (*Ontology Inference Layer*): Foi patrocinada por um consórcio da Comunidade Europeia por meio do projeto On-to-Knowledge. Gómez-Pérez (2002) a define como uma linguagem baseada em *frames* que utilizam lógica de descrição para fornecer uma semântica clara, ao mesmo tempo permitindo implementações eficientes de mecanismos de inferência que garantam a consistência da linguagem;
- DAML (*DARPA Agent Markup Language*): Segundo Breitman (2006), na mesma época que a o consórcio europeu estava criando a OIL, o *Defense Advanced Research Projects Agency* (DARPA), agência americana, em conjunto com o W3C, estava desenvolvendo a linguagem DAML. Essa linguagem foi criada por meio da extensão do RDF, de modo a acrescentar construtos mais expressivos, com objetivo de facilitar a interação de agentes de softwares autônomos na web;
- DAML+OIL: Foi criada como combinação das duas linguagens já citadas: DAML e OIL, uma vez que essas apresentavam características similares. É dividida em duas partes, como destaca Breitman (2006): a) Domínio dos objetos: consiste nos objetos que são membros de classes definidas na ontologia DAML; b) Domínio dos tipos de dados: consiste nos valores importados do modelo XML. Para a autora, a separação permite a implementação de mecanismos de inferência, uma vez que realizar inferências sobre tipos concretos de dados não seria possível;
- OWL (*Web Ontology Language*): O W3C lançou a OWL como uma revisão da linguagem DAML+OIL, e a projetou para atender às necessidades das aplicações para a web semântica, como Breitman (2005) menciona: a) construir ontologias para criar ontologias e explicitar conceitos e propriedades fornecendo informações sobre eles; b) explicitar fatos sobre um determinado domínio, fornecendo informações sobre indivíduos que fazem parte do domínio em questão; c) racionalizar sobre ontologias e fatos, determinando as consequências do que foi construído e explicitado.

2.2.4 Ferramenta para construção da Ontologia

Existem várias ferramentas para construção de ontologias. Uma ontologia pode ser desenvolvida em um editor de texto puro, escrevendo-se o código como se desenvolve um programa, mas o uso de ferramentas que auxiliem o processo de construção de ontologias contribui na agilidade de desenvolvimento, minimizando erros. (SANTARÉM, 2010).

Alguns exemplos de ferramentas para a construção de ontologias descrita pelo autor supracitado são:

- OilEed: editor simples, freeware, que suporta OWL, mas apresenta limitações para construção de ontologias em larga escala;
- OntoEdit: ambiente gráfico para desenvolvimento e edição de ontologias que segue os padrões do W3C. Permite inspeção, codificação, navegação e alteração de ontologias;
- Chimaera: ferramenta de diagnóstico de ontologias, que pode ser utilizada como auxiliar no desenvolvimento de ontologias para verificação de sintaxe, comparação de ontologias e para indicar classes e atributos semelhantes;
- API Jena: é um framework desenvolvido nos laboratórios da HP que permite o desenvolvimento e manipulação de ontologias por meio de linguagens de programação como o JAVA, ou seja, que utilizam linguagem orientada a objetos;
- Protégé: ambiente de código aberto (*open source*) para criação e edição de ontologias e bases do conhecimento. Destaca-se entre as demais ferramentas *open source* para desenvolvimento e manipulação de ontologias pela clara apresentação visual e facilidade de operação para usuários que não têm muita experiência.

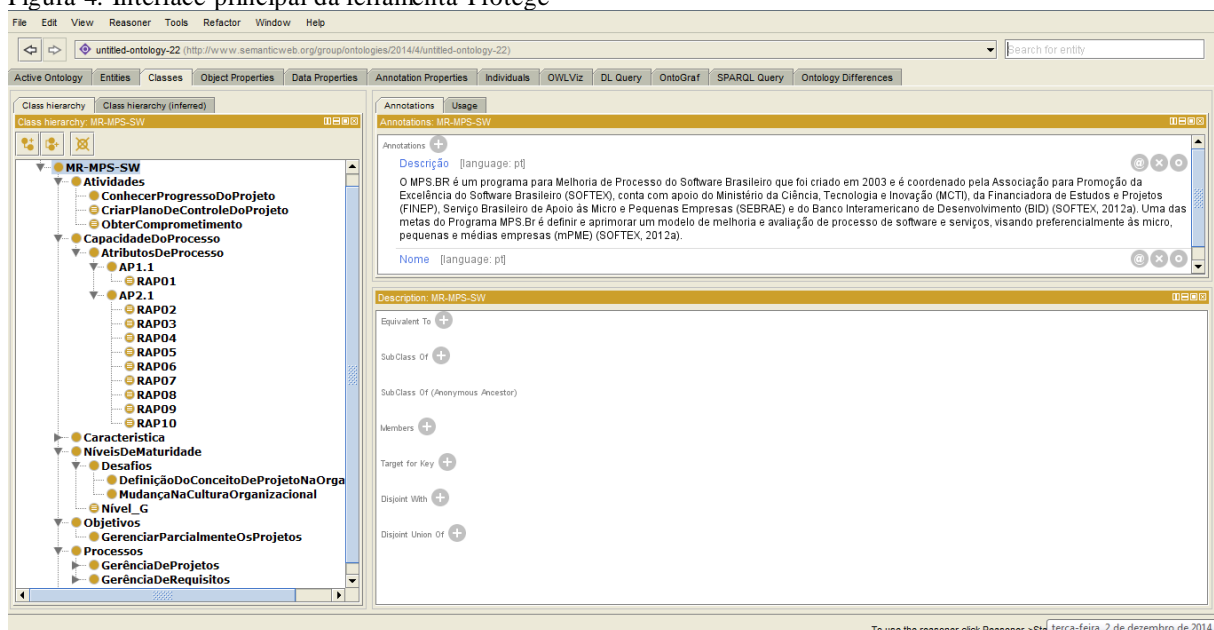
Dentre as ferramentas para a criação de ontologias, o Protégé merece destaque por ser a mais difundida no meio e escolhida como ferramenta para elaboração da ontologia proposta.

O Protégé é um framework Java, de uso livre e com código-fonte aberto (*open source*), cuja arquitetura extensível permite gerar e personalizar a criação de bases de conhecimento (STANFORD, 2012). O sistema Protégé proporciona um ambiente integrado de edição e armazenamento de uma base de conhecimento sobre determinado domínio. Além de suporte à construção de ontologias de domínio e personalização de formas de aquisição de conhecimento, pode combinar/integrar ontologias existentes. Permite a extensão de objetos gráficos de interface para tabelas, diagramas e componentes, a fim de acessar outros sistemas baseados em conhecimento embutidos em aplicações. Também disponibiliza uma biblioteca

para uso de outras aplicações, de modo que acessem e visualizem suas bases de conhecimento (NOY; MCGUINNESS, 2006).

A construção de ontologias usando o Protégé é feita a partir da criação de uma hierarquia de conceitos ou classes, que podem ser concretas ou abstratas, dependendo da permissão de instanciação dos mesmos. Além da definição dos conceitos/classes, é possível definir propriedades e relações entre eles (NOY; MCGUINNESS, 2006). A Figura 4 apresenta a interface da ferramenta Protégé, com algumas das suas opções de visualização selecionadas

Figura 4: Interface principal da ferramenta Protégé



Fonte: Elaborado pelo autor

3 TRABALHOS CORRELATOS

Nas revisões de literatura, constatou-se que poucos trabalhos tem tratado de ontologias direcionadas ao domínio MPS.Br. e um número considerável de trabalhos que abordam o domínio de Engenharia de software com foco nos processos de desenvolvimento de software. Bertollo e Falbo (2005) propõem uma ontologia de Processos de Software, que foi graficamente representada utilizando UML e tem como objetivo fornecer a compreensão, por parte das organizações de desenvolvimento de software, dos conceitos relacionados com processos de software incluindo padrões e metodologias. Souza (2012) em seu trabalho aborda o uso de Ontologias na Garantia da Qualidade e Melhoria de Processos, onde é

apresentado uma visão geral do estudo da arte das ontologias e seu uso na garantia da qualidade e na melhoria de processos de software. O trabalho de Cunha (2011) não está relacionado a construção de ontologias. Através de um estudo de caso, Cunha (2011) apresenta as Dificuldades Encontradas na Implementação do nível G do MPS.Br em uma organização e explora as dificuldades encontradas durante a implementação desse nível em algumas empresas, onde são registradas as dificuldades, vantagens e lições aprendidas durante sua implementação.

Dentre os trabalhos correlatos, o que mais se aproxima do trabalho proposto, é o de Pizzoleto (2013), que apresenta uma ontologia empresarial aplicada aos níveis G e F do MPS-SW. Para construção da ontologia, Pizzoleto (2013) faz uma análise das estruturas dos treze guias do modelo MPS-SW, com o propósito de encontrar uma estrutura comum entre eles. Desenvolve também, uma metodologia baseada em ontologia empresarial que define o processo de mapeamento das informações contidas nesses guias e através de um levantamento, evidencia que alguns conceitos mencionados nos guias do MPS-SW poderiam ser bem mais explicados considerando a inserção de conceitos e terminologias contemplados no Guia de Conhecimentos em Gerenciamento de Projetos ou PMBOK (Project Management Body of Knowledge), como complemento ao modelo MPS-SW, por evidenciar que muitos dos termos contidos no MPS-SW, também estão contidos no PMBOK.

Além do PMBOK, Pizzoleto (2013) considera a inclusão de indicadores do modelo BSC (Balanced Scorecard), um modelo de mensuração de desempenho empresarial para complementar o grupo de indicadores do modelo MPS-SW. Sua ontologia, também incluiu um conjunto de testes com potenciais usuários do MPS-SW, com o propósito de verificar a usabilidade da mesma. A ontologia proposta por Pizzoleto (2013), foi construída em linguagem OWL e é voltada a organizações desenvolvedoras de software, visando simplificar e uniformizar a compreensão dos Níveis G e F do MPS-SW, onde os processos do nível G estabelecem mecanismos para serem usados em processos gerenciais críticos no desenvolvimento de software: gerência de projetos e de requisitos e os do nível F, processos de apoio que asseguram a qualidade dos produtos e do processo, bem como gerenciam as configurações dos produtos.

Apesar de toda a abrangência contida na ontologia de Pizzoleto (2013), observa-se em seu trabalho, ausência das informações adicionais contidas nos guias Parte 1: Fundamentação para Implementação do Nível G; Parte 8: Implementação do MR-MPS em organizações que Adquirem Software e Parte 10: Implementação do MR-MPS em organizações do tipo Fábrica de Teste, para os resultados dos GPRs e GREs, pois de acordo com os esses guias, a área de

Gerência de Projetos (GPR), possui dezenove (19) GPRs, conforme Tabela 4 e a área de Gerência de Requisitos (GRE), possui cinco (05) GREs, conforme Tabela 5, aos quais, cada um tem um propósito geral e de acordo com o tipo de organização, há comentários adicionais que devem ser levados em consideração quanto ao tipo de organização. No que se refere ao Guia Parte 9: Implementação do MR-MPS em organizações do tipo Fábrica de Software, algumas informações adicionais destinadas a esse tipo de organização são consideradas no trabalho de Pizzoleto (2013) e apresentadas como resultados esperados para alguns GPRs e GREs.

Embora não exista uma regra obrigatória para determinar o nome das classes e propriedades em OWL, existem recomendações de guia prático de construção de ontologias OWL, Horridge (2008) para definição desses nomes. Observa-se que a ontologia proposta por Pizzoleto não segue essas recomendações. Dessa forma, o trabalho proposto buscou seguir essas recomendações destinadas aos nomes de classes e propriedades, pelo fato dessas recomendações ajudarem a facilitar o entendimento sobre as propriedades e serem importante para consistência da ontologia, como também, por permitem usar a English Prose TooltipGenerator (ferramenta geradora de dicas em inglês), a qual se utiliza dessa convenção para gerar expressões legíveis e descrever classes. A ferramenta exibe a descrição da classe em linguagem natural (inglês), facilitando o entendimento de uma descrição de classe. Ela é ativada quando o cursor é colocado sobre a descrição da classe na interface do usuário.

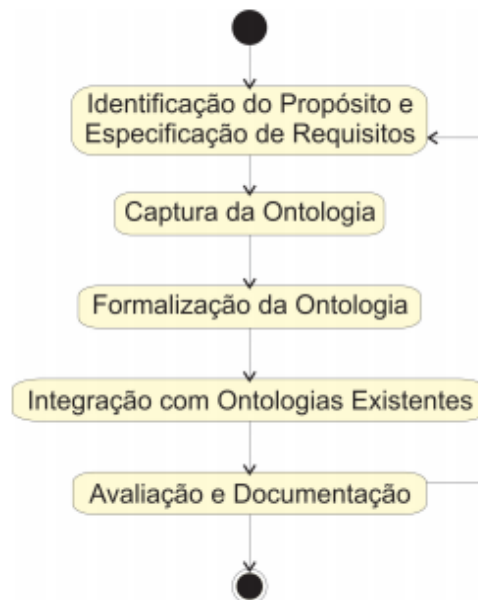
3.1 DESENVOLVIMENTO DA ONTOLOGIA

Esta seção descreve a metodologia utilizada para construção da ontologia, seu propósito, suas questões de competências e seus potenciais usuários, tomando como base o do Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS-SW:2012 de setembro de 2013 e os Guias Parte 08, Parte 09 e Parte 10, respectivamente, direcionados a Empresas que adquirem software, Fábrica de Software e Fábrica de Teste, publicados em novembro de 2011. Para a construção da ontologia, foram usados os guias citados e trabalhos relacionados. Com base nos Guias, foram definidas as especificações e identificadas as Classes, relacionamentos e os demais elementos necessário para a construção da ontologia.

4 PROCEDIMENTOS METODOLÓGICOS

Para a construção de uma ontologia, é importante seguir métodos apropriados que facilitam e orientam o seu desenvolvimento. A metodologia utilizada para o desenvolvimento deste trabalho, seguiu o método SABiO (Systematic Approach for Building Ontologies) proposto por (Falbo, 1998), pois através dessa metodologia, Falbo (1998), reuniu as melhores características das metodologia para construção de ontologia propostas por Gruninger; Fox (1995), Uschold; Gruninger, 1996 e Fernández; Gómez-Pérez; Juristo (1997), conforme Figura 8, que sugere a realização das seguintes atividades:

Figura 5: Processo iterativo de construção da ontologia



Fonte: Falbo (1998)

(I) Identificação do propósito e especificação de requisitos, tendo como produto principal as questões de competência que a ontologia deve ser capaz de responder. (II) Captura da ontologia, onde são capturados conceitos, relações, propriedades e restrições relevantes sobre o domínio em questão. (III) Formalização, na qual os axiomas da ontologia são escritos em uma linguagem formal, neste trabalho optou-se pela lógica descritiva. (IV) Integração com ontologias existentes, com objetivo de reutilizar conceituações existentes e integrá-las à ontologia em desenvolvimento relacionada ao trabalho em questão; (V) Avaliação da ontologia, que busca avaliar se a ontologia é capaz de responder às questões de competência e (VI) Documentação da ontologia, que tem por objetivo registrar o desenvolvimento da ontologia.

Seguindo a metodologia proposta, o domínio escolhido para captura e representação através de ontologia foi o nível G do MPS.Br, tomando como base as estruturas de sete (07) guias, vide Tabela 1, respectivamente, Parte 1, 8, 9, 10, 11, 12 e 13, com o propósito de analisar estruturas comum entre eles, levando em consideração, referências contidas em artigos e publicações que abordam o MPS-SW

Identificação do propósito e especificação de requisitos,

Nesta etapa de construção da ontologia, foram, realizadas análises dos guias a serem considerados para construção da ontologia e tomou-se como base a estrutura de sete (07) guias, vide Tabela 1, respectivamente, Parte 1, 8, 9, 10, 11, 12 e 13, além de trabalhos relacionados a mesma. Analisados os guias, foram identificadas suas questões de competência como: Quais são os processos envolvidos no Nível G?; Quais as dependências entre eles? ; Quais os resultado esperados de cada processo?; Quais os resultados esperados dos atributos de processo?; Quais os resultados esperados para Gerência de Requisitos?; Quais os resultados esperados para uma organização do tipo Adquirentes de Software?; Quais os resultados esperados para uma organização do tipo Fábrica de Software?; Quais os resultados esperados para uma organização do tipo Fábrica de Teste?; dentre outros. Após definição de suas questões de competências, foi definido seu uso como uma ferramenta de apoio que busca ajudar as pessoas a compreenderem melhor os passos a serem seguidos durante a implantação do nível G, com o propósito de organizar, simplificar, apoiar e facilitar a compreensão dos processos envolvidos nesse nível do guia Melhoria de Processo do Software Brasileiro (MPS-SW), tendo como potenciais usuários, empresas desenvolvedoras de software que buscam certificação no nível G, além de avaliadores e pessoas envolvidas nesse processo.

Captura da ontologia

Após identificação do propósito e especificações da Ontologia, a fase de captura consistiu no estudo do domínio contido no Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS-SW:2012, de setembro de 2013 e das partes 08, 09 e 10 do guia de implementação aplicado a organizações do tipo Adquirentes de Software, Fábrica de Software e Fábrica de Teste. Com base nas questões de competência definidas, foi realizada uma captura do conjunto de elementos contido nesse domínio que poderiam ser representados através de uma ontologia. A captura envolveu a identificação e especificações de conceitos (classes), seus relacionamentos e todos os demais elementos necessária para a representação da ontologia, tais como propriedades, axiomas, instâncias, etc.

Formalização da Ontologia

Definidos os elementos contidos no domínio e identificados as especificações de conceitos, o passo seguinte desse trabalho, consistiu em especificá-los. Na prática, uma ontologia pode ser representada através de qualquer linguagem não formal (natural), porém, a representação formal é considerada a mais apropriada. Neste trabalho, optou-se pela lógica descritiva na sua linguagem OWL, pois atualmente é a linguagem recomendada pelo W3C para pessoas que queiram construir ontologias, por possuir mais facilidades para expressar significados e semânticas do que outras linguagens como XML, RDF e RDF Schema. A OWL foi projetada para ser usada por aplicações que necessitem processar o conteúdo de informações, ao invés de somente apresentar a visualização destas informações.

Integração com ontologias existentes

De acordo com Falbo (2004), durante os processos de captura e/ou formalização, podem surgir a necessidade de integrar a ontologia em questão com outras já existentes, visando aproveitar conceituações previamente estabelecidas. No trabalho de Pizzoleto (2013), é proposto uma ontologia empresarial aplicada aos Níveis G e F do MPS.Br. Apesar de haver um trabalho semelhante, optou-se por não integrá-lo ao trabalho proposto, visto que o trabalho de Pizzoleto (2013) abrange dois Níveis do MPS.Br, e embora não existam convenções de nomenclatura obrigatórias para classes OWL, o trabalho de (Pizzoleto, 2012) não segue recomendação adequada para os nomes de classes e propriedades de uma ontologia criada em OWL que facilitam seu entendimento e a tornam mais consistente e o trabalho proposto buscou seguir essas recomendações. Seu trabalho também utiliza-se de conceitos do PMBOK e BSC para o desenvolvimento da ontologia, diferente do trabalhos proposto que utiliza-se apenas dos guias de implementação e trabalhos relacionados aos guias. Em seu trabalho, Pizzoleto (2013), não contempla informações adicionais contidas nos guias, que são destinadas as organizações do Tipo Adquirentes de Software, Fábrica de Software e Fábrica de Teste, o que o trabalho proposto buscou contemplar. Apesar da não integração, este trabalho optou por construir uma nova ontologia tomando como base o trabalho proposto por Pizzoleto, (2013) reutilizando conceituações existentes correlacionadas com ontologia desenvolvida, tomando como base os Guia de implementação do nível G do MPS-SW e os Guia de implementação do nível G em organizações do tipo: Adquirentes de Software, Fábrica de Software e Fábrica de Teste. O presente trabalho seguiu as recomendações para nomes de classes e propriedades em OWL, inseriu na ontologia, informações adicionais para organizações que adquirem software, Fábrica de Software e Fábrica de teste.

Avaliação e Documentação

A avaliação e documentação da ontologia foi realizada utilizando-se dos próprios recursos da ferramenta Protégé 4.3, a partir da inserção de anotações em seus componentes. A avaliação da ontologia, está descrita na seção 5.4 e foi realizada através do plug-in FaCT ++, que é um classificador de terminologias da ontologia usado para verificar se as declarações e as definições da ontologia são mutuamente consistentes entre si e reconhecer se os conceitos são adequados as definições e do Reasoner, um mecanismo de inferência utilizado para classificação automática das classes e verificação de consistência na ontologia.

5 CONSTRUÇÃO DA ONTOLOGIA

Para a construção da ontologia, inicialmente, foram analisados as estruturas de sete (07) guias, vide Tabela 1, respectivamente, Parte 1, 8, 9, 10, 11, 12 e 13, com o propósito de analisar estruturas comum entre eles. Partindo dessa análise, observou-se que os guias apresentam informações referentes à fundamentação teórica, propósito e necessidades. Percebe-se também, que em cada nível há diversos processos e cada um deles, tem sua capacidade. Além disso, cada um desses processos podem ter vários resultados. Em cada nível, há também, capacidades que são representadas por um conjunto de atributos descritos em termos de Resultados Esperados (RAP). Esses guias foram a base principal para o desenvolvimento da Ontologia, porém, a construção da mesma, também levou em consideração, referências contidas em artigos e publicações que abordam o MPS-SW.

Uma vez definidos estes elementos, optou-se por não seguir a metodologia baseada em ontologia empresarial utilizada por Pizzoleto (2013) e utilizar a metodologia SABiO proposta por Falbo (1998), pois através dessa metodologia, Falbo (1998), reuniu as melhores características das metodologia para construção de ontologia propostas por Gruninger; Fox (1995), Uschold; Gruninger, 1996 e Fernández; Gómez-Pérez; Juristo (1997), com as seguintes etapas: Propósito e Especificação de Requisitos, Definição do Propósito, Captura da Ontologia, Construção Formalização da Ontologia, Integração de Ontologias Existentes, Avaliação e Documentação da Ontologia

Com base nas pesquisas e após escolha do método, foi realizada uma pesquisa das ferramentas disponíveis para construção da ontologia e dentre as ferramentas existentes, optou-se pela escolha da Protégé V4.3, um editor de ontologias freeware e de código aberto desenvolvido na linguagem de programação java pelo departamento de informática médica da

Universidade de Stanford e utilizado para construir ontologias de domínio. Sua escolha se deu por ser considerada autoexplicativa, não requerendo treinamento para os futuros usuários. A construção da ontologia foi feita em linguagem OWL utilizando-se de três plug-ins do sistema Protégé:

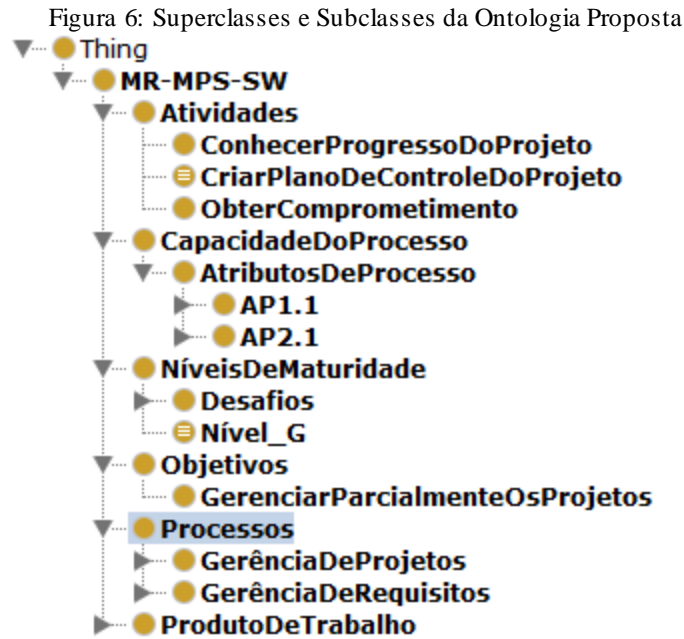
- OWLViz: Que permite visualizar as declarações e inferências obtidas. Ele também facilita a navegação gradativa entre as classes e permite visualizar e comparar as hierarquias das mesmas.
- OntoGraf, que permite visualizar e manipular os dados e informações presentes na ontologia. Ele tem um componente gráfico bem laborado, que mostra visualmente a agregação das classes.
- FaCT ++, que é um classificador de terminologias da ontologia usado para verificar se as declarações e as definições da ontologia são mutuamente consistentes entre si e reconhecer se os conceitos são adequados as definições.

5.1 CRIAÇÃO DAS CLASSES

Tendo em vista a grande quantidade de classes obtidas para a ontologia proposta e com o objetivo de facilitar o entendimento e a visualização dos procedimentos adotados, a apresentação da fase de criação das classes foi separada por grupos de classes. Mesmo com a utilização dessa estratégia, não serão apresentados todas as classes criadas na ontologia, apenas as principais. Mas não acreditamos que haverá prejuízo na compreensão do todo.

Toda nova ontologia contém a classe denominada owl: Thing. As classes são interpretadas como conjuntos de indivíduos (ou conjuntos de objetos). A classe owl: Thing é a classe que representa o conjunto que contém todos os indivíduos, uma vez que todas as classes são subclasses de owl: Thing, conforme Figura 8.

A ontologia que representa o domínio contido no Nível G, está dividida em seis (06) classes principais conforme Figura 8. As características dessas classes, estão descritas na Tabela 6. As principais características da ontologia, são consideradas como superclasses e as características associadas as essas superclasses, são consideradas subclasses, conforme Figura 8.



As classes OWL são conjuntos que contêm os indivíduos. Por exemplo a classe MR-MPS-SW, contém todos os indivíduos que fazem parte do Domínio MPS-SW no Nível G. Como se observa na Figura 6, as classes podem ser organizadas em hierarquias superclasse-subclasse, também conhecidas como taxonomias. A tabela abaixo, descreve as funções de algumas superclasses da Ontologia proposta.

Tabela 6 - Descrição das principais classes da Ontologia proposta

NOME DA CLASSE	DESCRIÇÃO
MR-MPS-SW	Esta classe está modelada na ontologia como sendo a classe mãe de todas as outras, pois essa classe representa o MPS-SW. Todas as subclasses dessa classe são tidas como subcaracterísticas do MPS-SW
ATIVIDADES	Corresponde as atividades envolvidas no Nível G do MPS-SW, ao qual possui como atividades: Criar o Plano do Projeto; Conhecer o Progresso do Projeto e Obter o Comprometimentos, que estão modeladas como subclasses da classe ATIVIDADES
CAPACIDADE DO PROCESSO	Esta classe representada um conjunto de atributos de processo descrito em termos de resultados esperados para o Nível G. Ela possui como subclasse, os resultados de todos os atributos de processos, conforme Tabela 3.
NÍVEIS DE MATURIDADE	Classe que corresponde aos níveis de maturidade presentes no MPS-SW. Essa classe, contém a subclasse nível G com suas características e seus desafios.
OBJETIVOS	Classe que corresponde aos objetivos de cada nível de maturidade contidos no MPS.Br. Por estar sendo modelado apenas o nível G do MPS-SW, está classe possui como subclasse a principal característica desse nível: Gerenciar Parcialmente o Projeto.

PROCESSOS

Esta classe representa todos os processos contidos no nível G. Ela possui como subclasse, todos os propósitos contidos na Gerência de Projetos, Tabela 4 e Gerencia de Requisitos, Tabela 5.

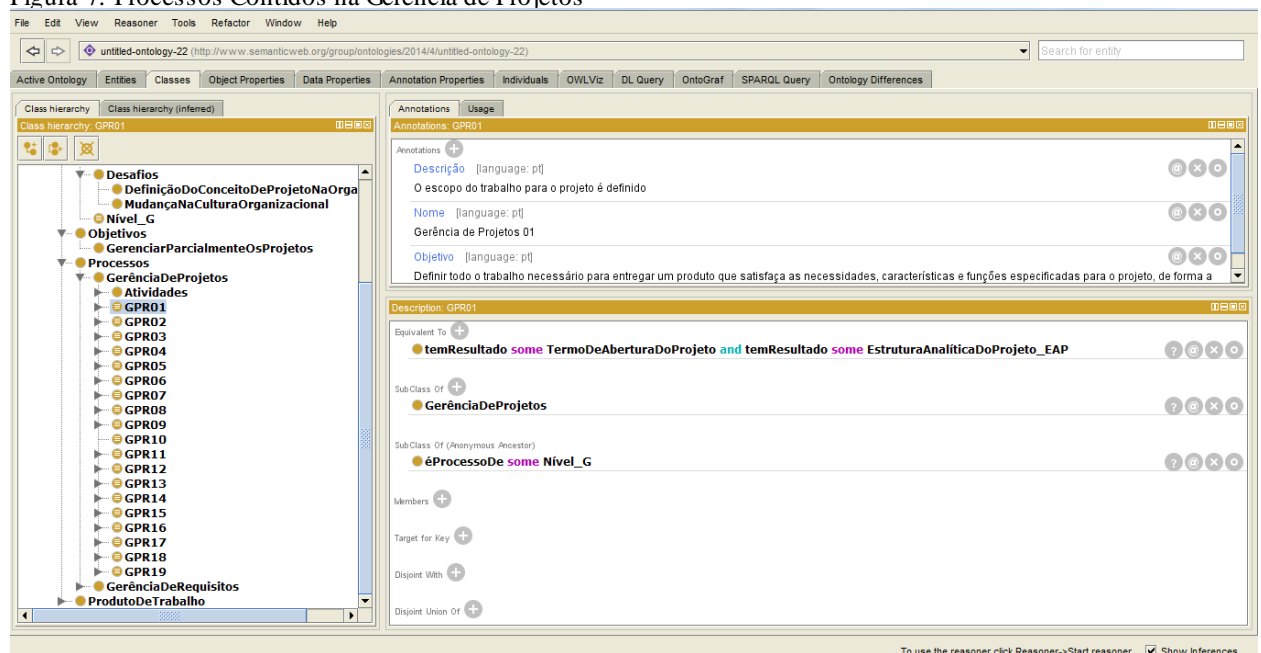
PRODUTO DE TRABALHO

Classe que representa todos os documentos que são gerados como resultados da execução dos processos como por exemplo: Estrutura Analítica do Projeto, que é resultante do GPR01, Ciclo de Vida do Projeto, que é resultante do GPR03, dentre outros.

Fonte: Elaborado pelo Autor

As subclasses são especializações de suas superclasses. Por exemplo, considere-se a Figura 7. A classe Processos e a Classe Gerência de Projetos. Gerência de Projetos é subclasse de Processos. Isso quer dizer que: Todos os processos contidos na Gerência de Projeto (GPRs) são Processos; Ser um GPR implica ser um Processo.

Figura 7: Processos Contidos na Gerência de Projetos



5.2 PROPRIEDADES DE OBJETO

As propriedades representam os relacionamentos existentes entre indivíduos de duas classes. Na ontologia proposta, foram criadas as seguintes propriedades:

NOME DA PROPRIEDADE DO OBJETO	DESCRIÇÃO
temAtividade	Esta propriedade relaciona as atividades envolvidas no Nível G
temCaracteristica	Esta propriedade relaciona as características envolvidas no Nível G
temDesafio	Esta propriedades relaciona os desafios envolvidos no Nível G
temObjetivo	Esta propriedade relaciona os objetivos presentes no Nível G
temObrigação	Esta propriedades relaciona as obrigações que envolve os processos presentes no Nível G

temProcesso	Esta propriedade relaciona as áreas de conhecimento aos processos de gerenciamento que compõem o Nível G
temResultado	Esta propriedades relaciona os resultados obtidos nos processos contidos no Nível G
éAtividadeDe	Esta propriedade é inversa a propriedade temAtividade e relaciona as atividades presentes no Nível G
éCaracterísticaDe	Esta propriedade é inversa a propriedade temCaracterística e relaciona as características envolvidas no Nível G
éDesafioDe	Esta propriedade é inversa a propriedade temDesafio e relaciona os presentes envolvidos no Nível G
éObjetivoDe	Esta propriedade é inversa a propriedade temObjetivo e relaciona os objetivos presentes no Nível G
éObrigaçãoDe	Esta propriedades é inversa a propriedade temObrigação e relaciona as obrigações que envolvem os processos presentes no Nível G
éProcessoDe	Esta propriedade é inversa a propriedade temProcesso e relaciona as áreas de conhecimento aos processos de gerenciamento que compõem o Nível G
éResultadoDe	Esta propriedades é inversa a propriedade temResultado e relaciona os resultados contidos nos processos do Nível G

Essas quatorze (14) propriedades criadas podem ser consideradas como sete (07) pares de propriedades, onde uma propriedade é inversa da outra, possibilitando o inter-relacionamento entre duas classes. A Figura 8 ilustra as propriedades criadas no Protégé e as respectivas propriedades inversas. A Figura 9 ilustra o inter-relacionamento entre duas classes da ontologia proposta.

Figura 8: Propriedades criadas para relacionar as classes e suas respectivas propriedades

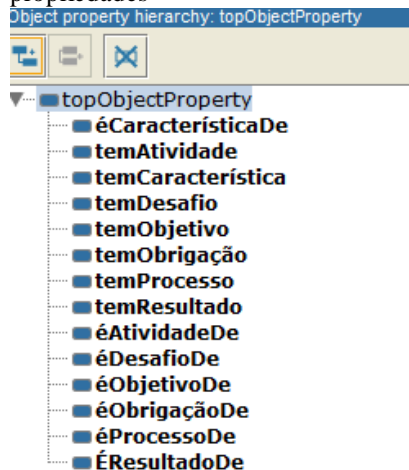
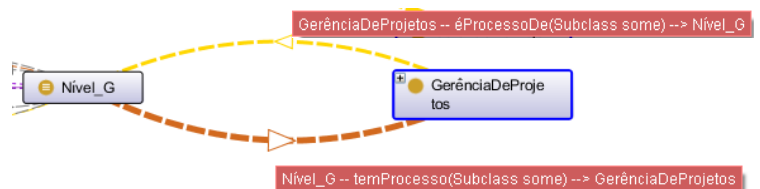
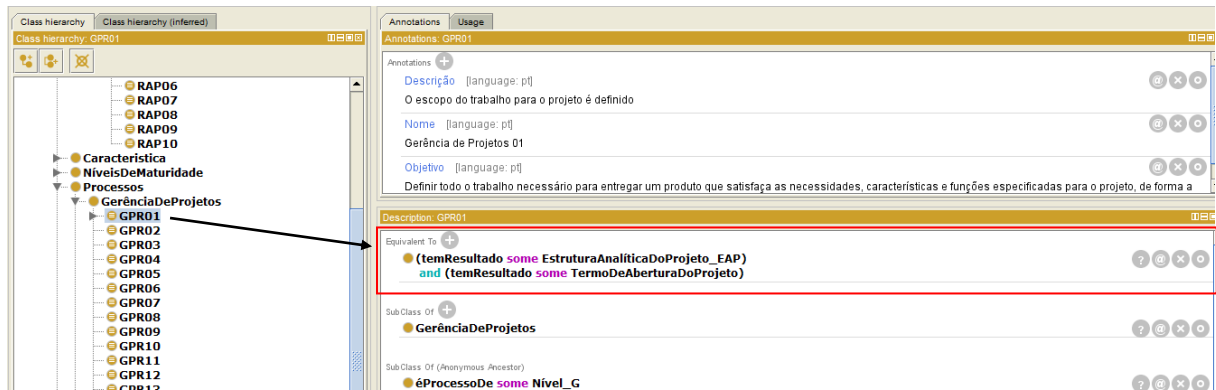


Figura 9: Inter-relacionamento entre duas classes, proporcionado pela propriedade inversa



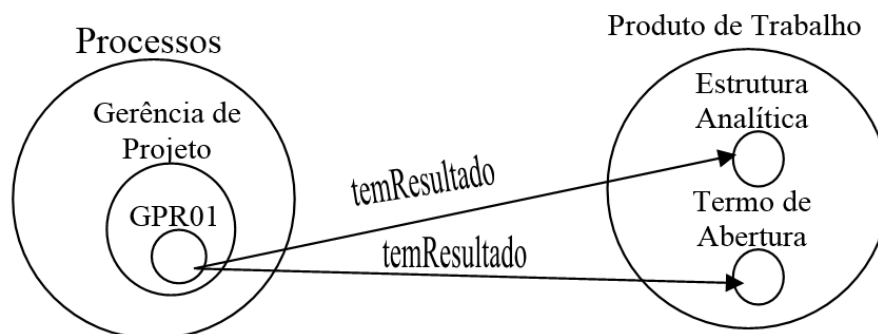
Entre as propriedades do objeto (objects properties) e as classes, as relações ocorrem de forma binárias e permitem inferir informações sobre as mesmas, conforme Figura 10.

Figura 10: Propriedade inferindo classe



Por exemplo, a propriedade `temResultado`, Figura 8, pode ligar a subclasse `GPR01`, que está contida na superclasse `Processos` a subclasse `TermoDeAberturaDoProjeto` e a subclasse `EstruturaAnaliticaDoProjeto_EAP`, ambas contidas na superclasse `ProdutosDeTrabalho` conforme Figura 10. Nesse sentido, da forma como está modelado, quando falamos de `GPR01`, estamos falando que este processo tem um Termo de Abertura do Projeto e uma Estrutura Analítica do Projeto, que estão mapeados na ontologia como uma relação de equivalência entre `GPR01` que tem como resultado, um Termo de Abertura do Projeto e uma Estrutura Analítica do Projeto, ou seja, ter um Termo de Abertura do Projeto e uma Estrutura Analítica, implica dizer ter o `GPR01`. Uma melhor maneira de visualizar essa inferência está na Figura 11.

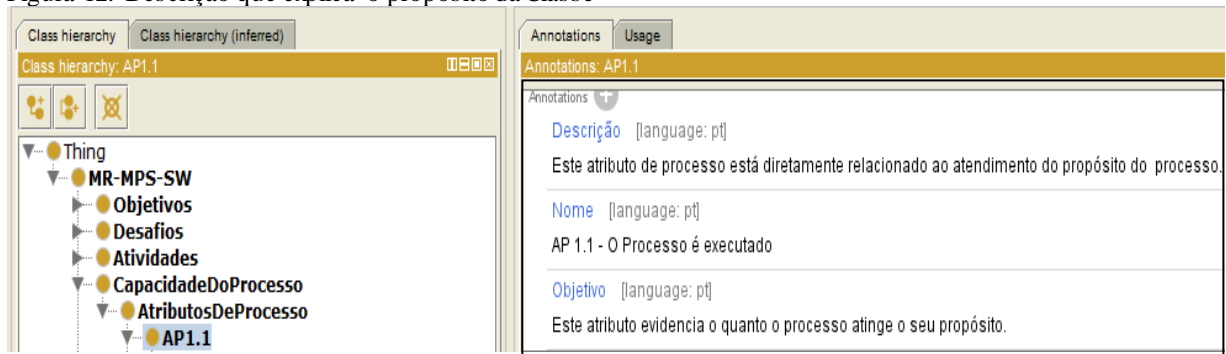
Figura 11: Representação de Propriedade inferindo classe



5.3 DESCRIÇÃO DAS CLASSES NA FERRAMENTA PROTÉGÉ

Com o propósito de facilitar o esclarecimento do usuário, à medida que ele navega pela ontologia, foram introduzidas nas classes, descrições que explicam o seu propósito, conforme Figura 12.

Figura 12: Descrição que explica o propósito da Classe



As Classes, Subclasses e os relacionamentos presentes na Ontologia proposta, foram definidos tomando como base o texto contido nos guias. Tomamos como exemplo, as Figuras 13 e 14, que apresenta trechos do guia Parte 1: Implementação no Nível G.

Figura 13: Texto Atributos de Processo no Nível G

7 Os atributos de processo no nível G

De acordo com o Guia Geral do MR-MPS-SW, “a capacidade do processo é representada por um conjunto de atributos de processo descrito em termos de resultados esperados. A capacidade do processo expressa o grau de refinamento e institucionalização com que o processo é executado na organização/unidade organizacional. No MR-MPS-SW, à medida que a organização/unidade organizacional evolui nos níveis de maturidade, um maior nível de capacidade para desempenhar o processo deve ser atingido” [SOFTEX, 2011a].

A seguir, os atributos de processo AP 1.1 e AP 2.1, conforme aplicáveis no nível G, são descritos com detalhes.

Fonte: (SOFTEX 2012a)

Figura 144: Descrição dos Atributos

7.1 AP 1.1 - O processo é executado

Este atributo evidencia o quanto o processo atinge o seu propósito.

Este atributo de processo está diretamente relacionado ao atendimento do propósito do processo. Relacionado a este atributo de processo está definido o seguinte resultado esperado:

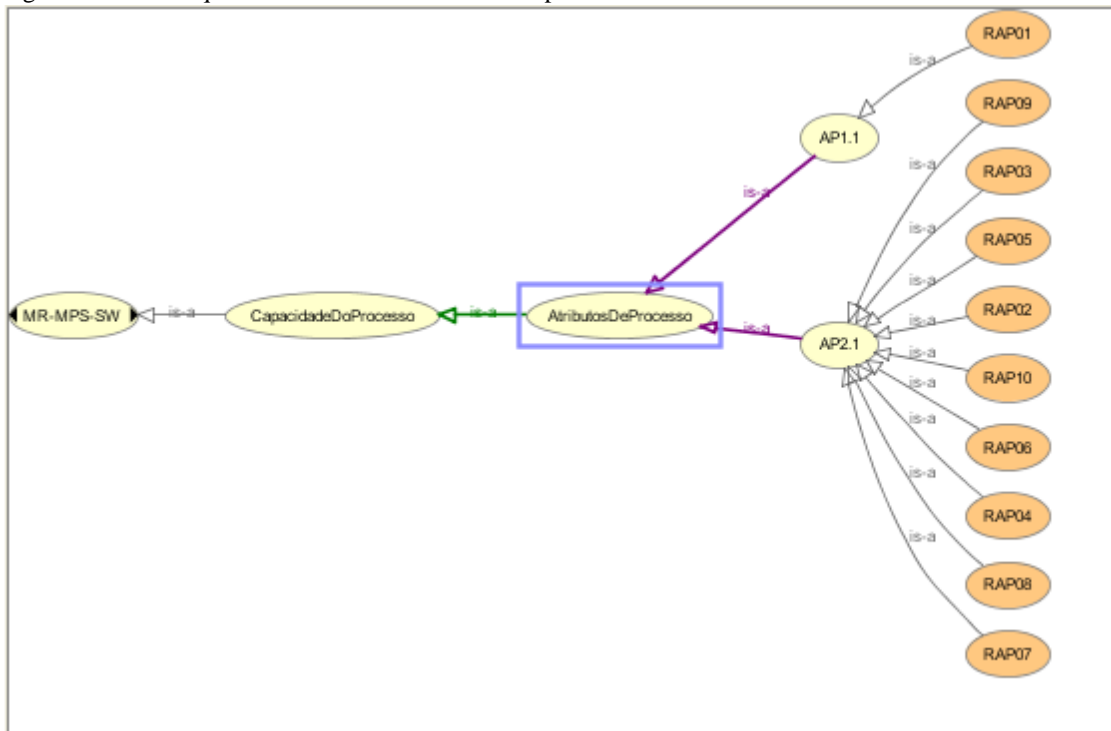
7.1.1 RAP1 - O processo atinge seus resultados definidos

Este resultado esperado busca garantir que o processo transforma produtos de trabalho de entrada identificáveis em produtos de trabalho de saída, também identificáveis, permitindo, assim, atingir o propósito do processo. Ou seja, este resultado implica diretamente na geração dos principais produtos requeridos pelos resultados dos processos.

Fonte: (SOFTEX 2012a)

O trecho do texto representado através das Figuras 13 e 14, corresponde ao item sete (07) do Guia, que trata dos atributos de processos no Nível G. De acordo com o texto das Figuras 13 e 14, nota-se que em meio a esses conceitos, existem palavras e termos chaves capazes de formar um domínio que represente e organize melhor o domínio desses processos. Em uma ontologia, as classes são representações concretas de conceitos. Elas são conjuntos que contêm os indivíduos, de forma que eles apresentem os requisitos para a participação na classe. Para construção das classes da ontologia proposta, foram consideradas palavras chaves contidas nos guias que em resumo, definem o domínio ao qual elas especificam. Por exemplo, para modelar os atributos dos processos contidos nessa parte do Guia, Figuras 13 e 14, foram consideradas como palavras chaves para formação das classes: Capacidades do Processo, por representar um conjunto de atributos de processos, Atributos de processos, que contém todos os atributos dos processos, os AP1.1 e AP2.1, que contém os atributos específicos a cada processo e os RAPs, que são os resultados esperados de cada atributo. Através da Figura 15, é possível visualizar a hierarquia das classes de como ficou a modelagem desses processos na ferramenta Protégé.

Figura 155: Hierarquia das Classes do Domínio Capacidade do Processo



Em OWL as classes são conjuntos que contêm os indivíduos. Por exemplo, a classe AtributosDeProcessos pode conter todos os indivíduos que são processos no domínio de interesse. Subclasses são especializações de suas superclasses. Por exemplo, a classe CapacidadeDoProcesso por conter o conjunto de atributos de processo, foi definida como superclasse AtributosDeProcesso, e assim AtributosDeProcesso, passou a ser subclasse de CapacidadeDoProcesso. Isso quer dizer que: Todos os AtributosDeProcesso são CapacidadeDoProcesso; Todos os membros da classe AtributosDeProcesso são membros da classe CapacidadeDoProcesso; Ser um AtributoDeProcesso implica ser uma CapacidadeDoProcesso.

5.4 VERIFICAÇÃO DA ONTOLOGIA

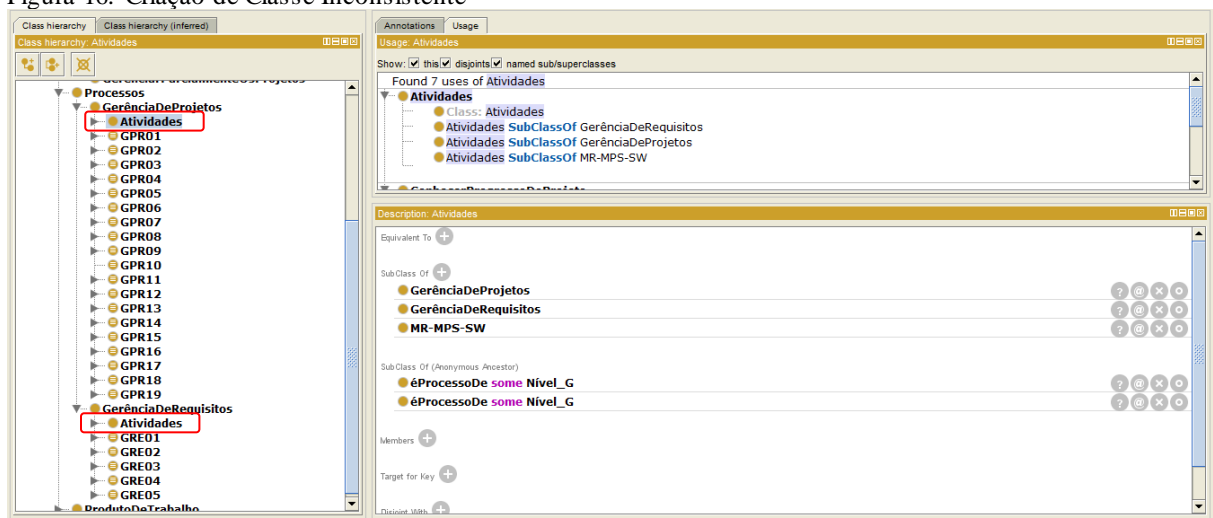
Após a criação das classes e propriedades, o mecanismo de inferência denominado Reasoner foi utilizado para classificação automática das classes e verificação de consistência na ontologia e do plug-in FaCT ++, usado para verificar se as declarações e as definições da ontologia são mutuamente consistentes entre si e reconhecer se os conceitos são adequados as definições.

5.4.1 DEMONSTRAÇÃO DE DETECÇÃO DE INCONSISTÊNCIA

Antes de efetuar a verificação da ontologia construída, é demonstrado o funcionamento do Mecanismo de Inferência para classificação automática das classes e para detecção de inconsistências. Para isso, foi especificada uma condição indevida no processo de Gerencia de Projetos, conforme descrito a seguir:

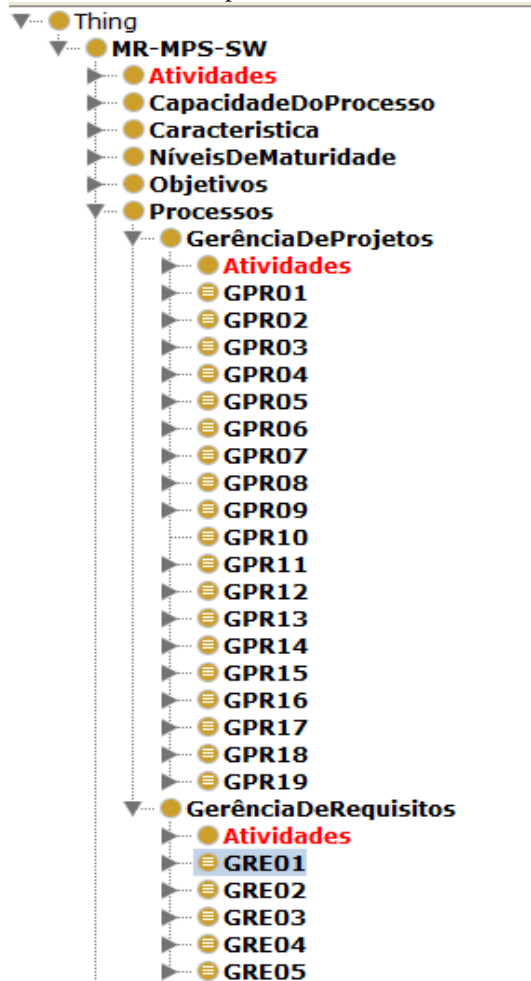
1. A classe Processos, contém as subclasses Gerência de Projetos e Gerência de Requisitos. Na classe Gerência de Projetos, existe uma subclasse com o nome atividades que descreve as principais atividades da Gerência de Projetos;
2. Apesar das Classes Gerência de Projetos e Gerência de Requisitos serem subclasses de uma classe maior, elas possuem objetivos e resultados diferentes, por isso, foram declaradas como disjunta, isso significa que uma subclasse só pode pertencer a uma única superclasse.
3. Propositamente, a subclasse atividades, que é subclasse da Gerência de Projetos, foi declarada como subclasse da Gerência de Requisitos, conforme Figura 16:

Figura 16: Criação de Classe Inconsistente



Após a execução da classificação automática das classes, o mecanismo de inferência aponta esta inconsistência, conforme a Figura 17:

Figura 17: A subclasse Atividades destacada como inconsistente pelo Reasoner

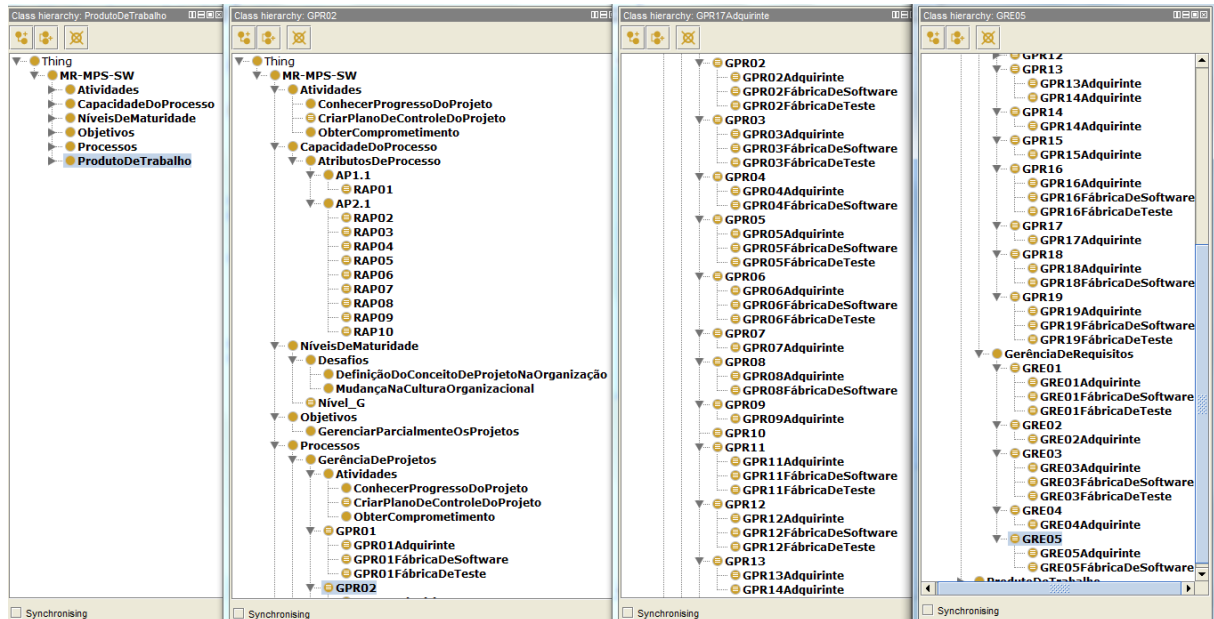


Adicionalmente, na janela de classes, a subclasse Atividades é destacada em vermelho, chamando a atenção para a inconsistência apurada, e aparece como sendo subclasse de Gerência de Projetos e Gerência de Requisitos, causada pela especificação incorreta das condições.

5.4.2 VERIFICAÇÃO DE INCONSISTÊNCIA NA ONTOLOGIA

Ao longo do desenvolvimento da ontologia proposta neste trabalho, a mesma foi analisada e revisada e as correções necessárias foram efetuadas ao longo de sua construção. O Reasoner presente na ferramenta Protégé, foi constantemente utilizado para verificar a existência de inconsistência na criação das classes e propriedades. O resultado obtido não apontou inconsistências, conforme demonstrado na Figura 18.

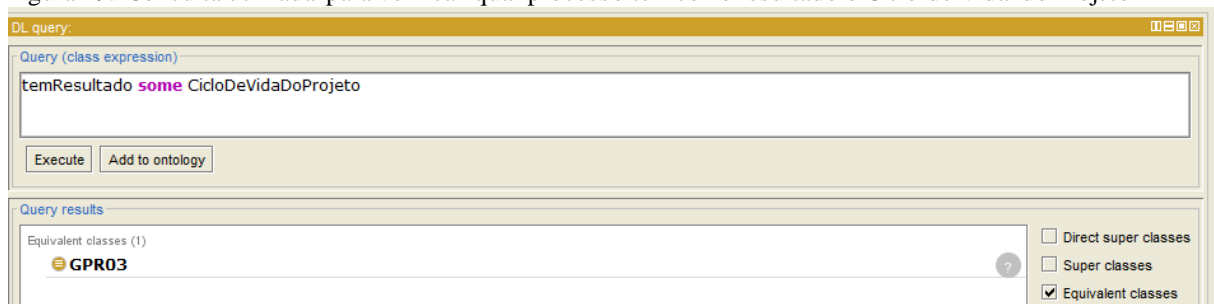
Figura 18: Visão parcial das classes construídas automaticamente pelo Reasoner sem apontamento de inconsistência



5.4.3 VERIFICANDO EQUIVALÊNCIAS ATRAVÉS DE CONSULTAS

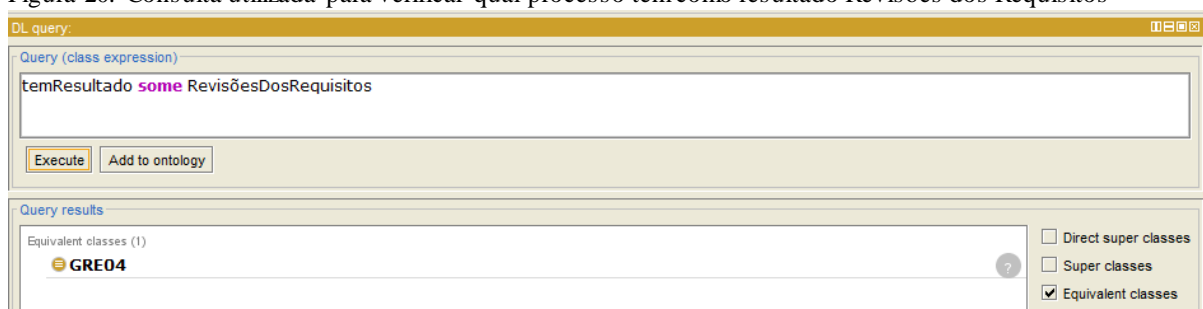
Para verificar a equivalência entre as classes, foram realizadas consultas através de uma ferramenta do Protégé chamada DL Query, com o propósito de verificar se as classes estavam de acordo com suas equivalências. As Figuras 19 e 20, mostram o exemplo de duas consultas utilizadas para verificar a quais processos pertencem respectivamente o Ciclo de vida do projeto e as Revisões de Requisitos.

Figura 19: Consulta utilizada para verificar qual processo tem como resultado o Ciclo de Vida do Projeto



A consulta representada através da Figura 19, foi realizada para saber qual processo equivale ao resultado o Ciclo de Vida do Projeto. Observa-se na Figura, que ao fazer essa consulta, a ferramenta apresenta como resultado o GPR03, ou seja, O GPR03, equivale a definição do Ciclo de Vida do Projeto.

Figura 20: Consulta utilizada para verificar qual processo tem como resultado Revisões dos Requisitos

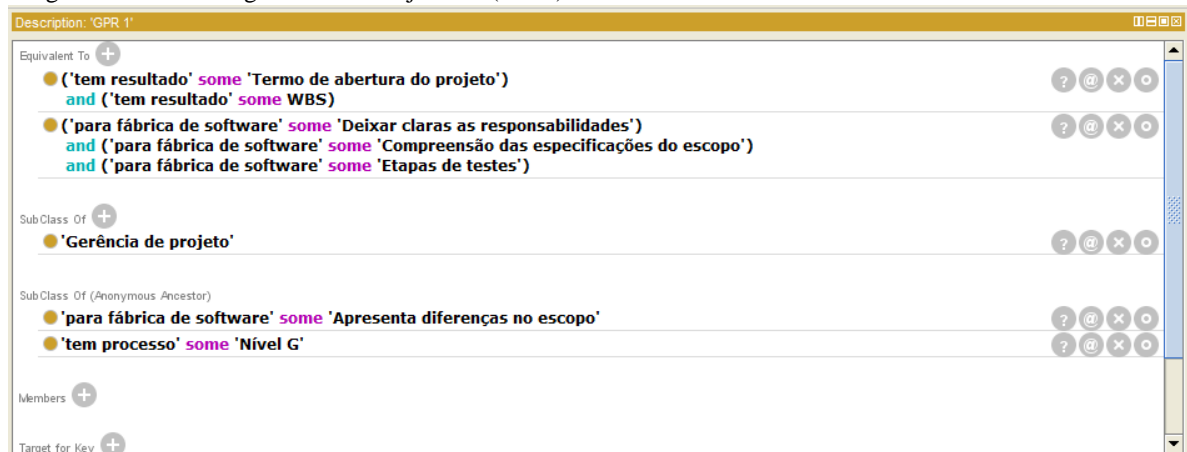


A consulta realizada na Figura 20, foi realizada para saber qual processo equivale ao resultado Revisões dos Requisitos. Observa-se na Figura, que ao fazer essa consulta, a ferramenta apresenta como resultado o GRE04, ou seja, O GPR04, no nível G, equivale a as Revisões de Requisitos.

5.5 DIFERENÇAS ENTRE OS TRABALHOS

Durante o processo de desenvolvimento da ontologia usou-se como referência a ontologia proposta por Pizzoleto (2013), entretanto, optou-se por não integra-la com a de Pizzoleto (2013), pois sua ontologia abrange dois Níveis do MPS.Br. Seu trabalho também utiliza-se de conceitos do PMBOK e BSC para o desenvolvimento da ontologia, diferente do trabalhos proposto que utiliza-se apenas dos guias de implementação e artigos relacionados ao MPS.Br e embora não exista regra obrigatória para nome de classes e propriedades, há recomendações que facilitam o entendimento dessas classes e tornam a ontologia mais consistente, o trabalho de Pizoleto não segue essas recomendações e a ontologia proposta, buscou segui-las. O trabalho proposto, também contempla informações adicionais nos guias direcionado a organizações do tipo Adquirentes de Software, Fábrica de Software e Fábrica de Teste e faz correções em algumas modelagens propostas por Pizzoleto (2013) que após análises, considera algumas modelagens inconsistentes, como por exemplo, na Figura 21, do trabalho de Pizoleto (2013), referente ao Processo Gerência de Projeto 01 - (GPR01), que tem como proposito: definir o escopo de trabalho para o projeto, tendo como resultado, o termo de abertura de projeto e a Work Breakdown Structure (WBS) ou Estrutura Analítica do Projeto, que identifica os produtos, serviços e resultados a serem feitos em um projeto, servindo como base para a maior parte do planejamento de projeto.

Figura 21: Processo gerência de Projeto 01 (GPR1)



Pizzoleto (2013)

Observa-se em sua na modelagem, que no GPR1, há um resultado geral, que é o termo de abertura do Projeto e a WBS, e outro para Fábrica de software, dando a ideia que o GPR1, possui dois resultados diferente. Na modelagem, não há referências adicionais para organizações do Tipo Adquirintes de Software e Fábrica de Teste conforme consta nos guias de implementação parte 1: Fundamentação para Implementação do Nível G.

Outro detalhe, trata-se do conceito de equivalência em OWL. Da forma como está a modelagem da Figura 21, o GPR1 possui dois resultado equivalentes: o primeiro, que de uma forma geral tem como resultado o Termo de Abertura do Projeto e a Work Breakdown Structure (WBS), e o segundo, que tem como resultado para um Fábrica de Software, deixar clara as responsabilidades, compreensão das especificações do escopo e etapas de teste.

Da forma como modelado, do ponto de vista lógico há inconsistências. Tomamos como exemplo a Figura 21, em que:

A = GPR01

B = ('tem resultado' **some** 'Termo de abertura do projeto') **and** ('tem resultado' **some** WBS)

C = ('para fábrica de software' **some** 'Deixar clara as responsabilidades') **and**
and ('para fábrica de software' **some** 'Compreensão das especificações do escopo')
and ('para fábrica de software **some** Etapas de teste)

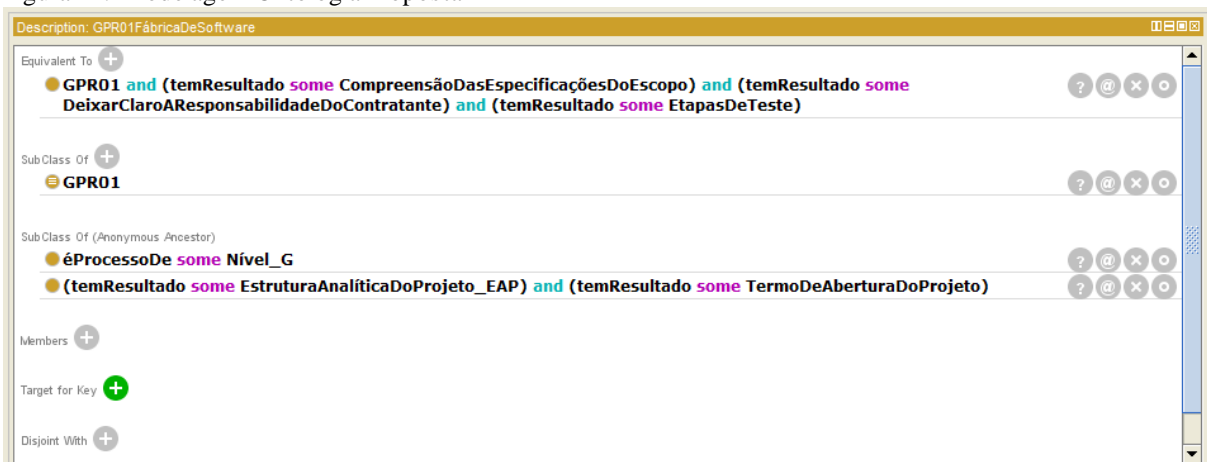
Logicamente, de acordo com a modelagem da Figura 21, podemos representa-la da seguinte maneira:

$$\begin{array}{l} A \leftrightarrow B \\ A \leftrightarrow C \end{array} \quad \Bigg| \quad B \leftrightarrow C$$

A equivalência $A \leftrightarrow B$, significa dizer que $A \rightarrow B$ e $B \rightarrow A$ onde a sentença correspondente a A implica em B , e sentença correspondente a B implica em A . A equivalência $A \leftrightarrow C$, significa dizer que $A \rightarrow C$ e $C \rightarrow A$, onde a sentença correspondente a A implica em C , e a sentença correspondente a C implica em A . Disso, conclui-se que os conceito B e C , deveriam ser equivalentes, porém, isso não ocorre e se dá pelo fato da modelagem em assumir que A equivale a B e A equivale a C , o que implicaria em B e C serem equivalentes, porém, B e C são conceitos disjuntos. B refere-se ao resultado geral esperado para GPR1, enquanto C , para um resultado específico de GPR1 para uma organização do tipo Fábrica de Software.

No entendimento deste trabalho, uma forma adequada para representar essa modelagem, está de acordo com a Figura 22, que elimina a inconsistência que foi inserida na modelagem apresentada na Figura 21.

Figura 22: Modelagem Ontologia Proposta



Fonte: Elaborado pelo autor

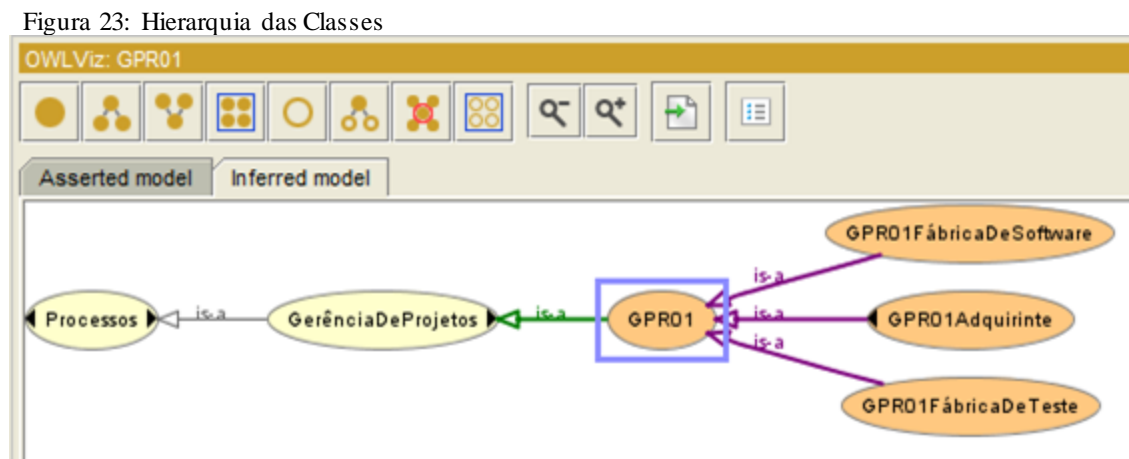
Em OWL, quando você cria uma subclasse, ela herda as propriedades de sua classe superior. Na modelagem representada na Figura 6, criou-se na classe GPR01, uma subclasse com o nome GPR01FabricaDeSoftware, ao qual, tem como resultado: Compreensão das especificações do Escopo, Etapas de Teste e Deixar clara as responsabilidades do contratante.

A representação lógica para a Figura 18 pode ser representada da seguinte forma:

$$A \leftrightarrow (B \wedge C) \not\leftrightarrow B \leftrightarrow C$$

De acordo com a representação acima, O GPR1 (A), quando se tratar de uma organização do tipo Fábrica de Software, equivale as sentenças B e C, com C complementando B, ou seja, para uma Fábrica de Software, GPR01 equivale ao Termo de Abertura do Projeto e a Estrutura Analítica do Projeto, e dentro desses resultados, há os resultados específicos que ela deve especificar: Compreensão das especificações do Escopo, Etapas de Teste e Deixar clara as responsabilidades do contratante.

Uma melhor maneira de entender como fica o relacionamento entre essas classes, é através do OWLViz, que permite a visualização a comparação das hierarquias das classes, conforme Figura 23.

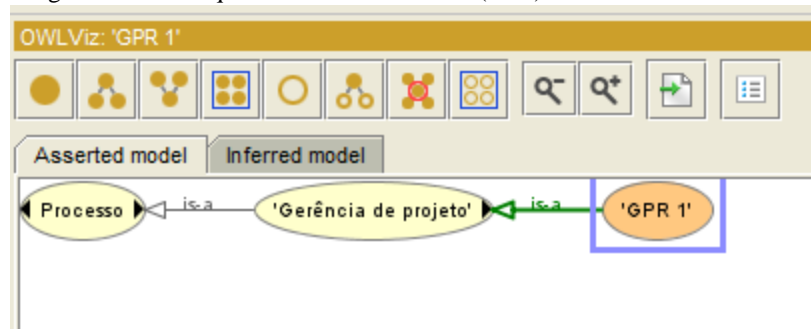


Fonte: Elaborado pelo autor

Conforme se observa na Figura 23 da ontologia proposta, a classe GPR01, passa ter três (03) subclasses, sendo elas: GPR01FabricaDeSoftware, GPR01Adquirintes e GPR01FábricaDeTeste, respectivamente, com elas contendo informações adicionais referente aos resultados do GPR01 direcionado a cada tipo de organização.

Na modelagem proposta por Pizzoleto (2013), usando o OWLViz não é possível visualizar subclasses no GPR1, pois em sua modelagem, o GPR1 não possui informações para Adquirintes de Software, Fábrica de Teste e algumas referências para Fábrica de Software são referenciadas usando uma objects properties com o nome: 'para fábrica de software', conforme Figura 21. Na Figura 24, há a hierarquia da classe GPR1 na ontologia de Pizzoleto.

Figura 24: Hierarquia do GPR1 Pizzoleto (2013)



A correção de equivalências efetuada para GPR01, conforme Figura 22, também foi aplicada aos demais processos contidos na Gerência de Projetos e Gerência de Requisitos que estavam modelados de forma semelhantes ao exemplo da Figura 21. Outra modelagem inconsistente que observa-se no trabalho de Pizzoleto (2013), está na definição do Nível G. na modelagem de Pizzoleto (2013), está de acordo com a Figura 25.

Figura 25: Definição do Nível G do Pizzoleto (2013)

The screenshot shows the OWL editor interface for the 'Nível G' class. On the left, a class hierarchy tree shows 'Nível G' as a subclass of 'Nível F'. The main area displays the class description for 'Nível G'. It includes annotations for 'nome' (Nível G), 'descrição' (Parcialmente Gerenciado), and 'identificador' (8.2.0.0). The 'Equivalent To' section lists five conditions:

- 'tem obrigação' some 'Registrar adaptação'
- ('tem desafio' some 'Definição de conceito') and ('tem desafio' some 'Mudança organizacional')
- ('tem processo' some 'Gerência de projeto') and ('tem processo' some 'Gerência de requisitos')
- ('tem característica' some 'AP 1.1') and ('tem característica' some 'AP 2.1')
- 'tem objetivo' some 'Gerenciar parcialmente os projetos'

 The 'Sub-Class Of' section lists two classes: 'Nível F' and 'Nível'.

Da forma como modelado, o Nível G possui cinco (05) equivalências e assim como ocorre na Figura 21, do ponto de vista lógico há inconsistências. Vejamos:

A: Nível G;

B: afirma que ele 'tem obrigação' **some** 'Registrar adaptação';

C: 'tem desafio' **some** 'Definição de conceito' **and** 'tem desafio' **some** 'Mudança organizacional'

D: 'tem processo' **some** 'Gerência de projeto' **and** 'tem processo' **some** 'Gerência de requisitos'

E: 'tem característica' **some** 'AP 1.1' **and** 'tem característica' **some** 'AP 2.1'

F: 'tem objetivo' **some** 'Gerenciar parcialmente os projetos'

De acordo com a modelagem da Figura 25, ao representar por exemplo os itens A, B e C, temos que:

$$\begin{array}{l} A \leftrightarrow B \\ A \leftrightarrow C \end{array} \quad \Bigg| \quad B \leftrightarrow C$$

A equivalência $A \leftrightarrow B$, significa dizer que:

- A (Nível G) \rightarrow B (Registrar adaptação);
- B (Registrar Adaptação) \rightarrow A (Nível G).

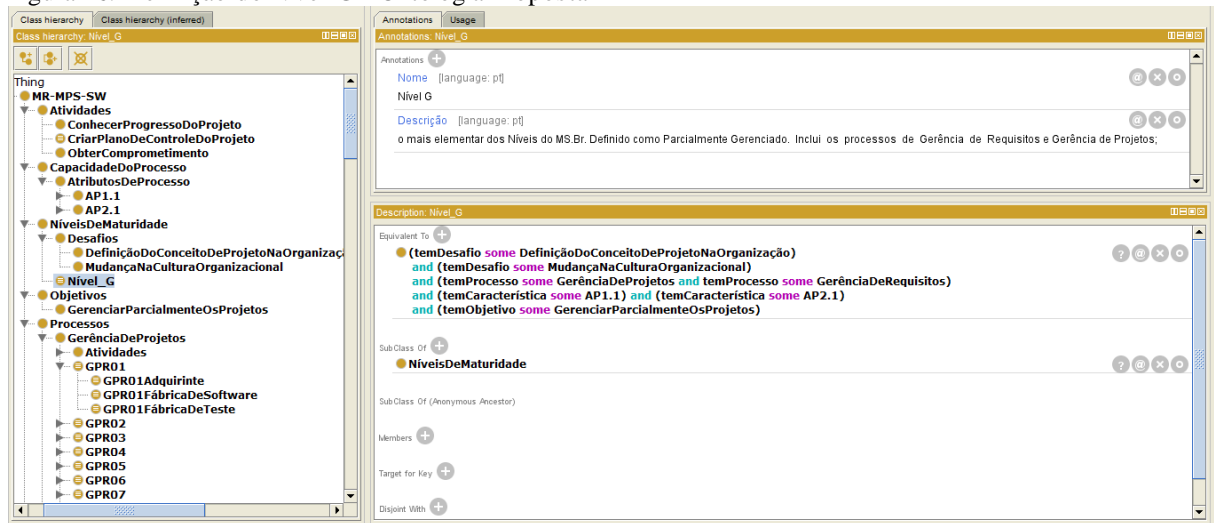
A equivalência $A \leftrightarrow C$, significa dizer que:

- A (Nível G) \rightarrow C (Definição de conceito e Mudança organizacional)
- C (Definição de conceito e Mudança organizacional) \rightarrow A (Nível G).

Disso, conclui-se que os conceitos: B (Registrar adaptação) e C (Definição de conceito e Mudança organizacional), deveriam ser equivalentes, porém, isso não ocorre e se dá pelo fato da modelagem em assumir que A (Nível G) equivale a B (Registrar Adaptação) e A (Nível G) equivale a C (Definição de conceito e Mudança organizacional), o que implicaria em B (Registrar Adaptação) e C (Definição de conceito e Mudança organizacional) serem equivalentes, porém, B e C são conceitos disjuntos. B refere-se a uma obrigação do Nível G, enquanto C, ao desafios do Nível G. Através dessas três (03) sentenças, é possível verificar uma inconsistência na modelagem representada através da Figura 25.

No entendimento deste trabalho, uma forma adequada para representar essa modelagem, está de acordo com a Figura 26, que elimina a inconsistência apresentada na Figura 25.

Figura 26: Definição do Nível G - Ontologia Proposta



De acordo com a representação acima, temos que:

A: Nível G;

B: 'temDesafio' some 'DefiniçãoDoConceitoDeProjetoNaOrganização' and 'temDesafio' some 'MudançaNaCulturaOrganizacional';

C: 'temProcesso' some 'GerênciaDeProjeto' and 'temProcesso' some 'GerênciaDeRequisitos'

D: 'temCaracterística' some 'AP 1.1' and 'temCaracterística' some 'AP 2.1'

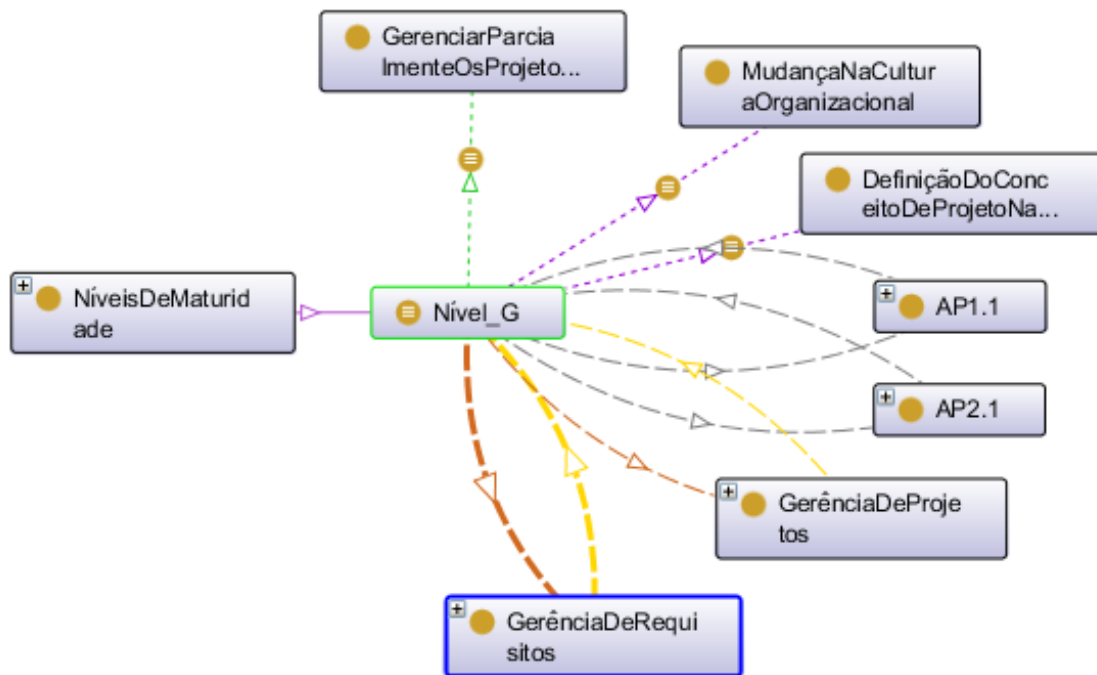
E: 'tem objetivo' some 'Gerenciar parcialmente os projetos'

A representação lógica para a Figura 26 pode ser representada da seguinte forma:

$$A \leftrightarrow (B \wedge C \wedge D \wedge E)$$

De acordo com a representação acima, ao definir o Nível G equivalente a união entre as sentenças B, C, D e E, elimina-se as inconsistências presentes na Figura 25, por tê-lo equivalente a união entre seus desafios: Definição do Conceito de Projeto na Organização e Mudança na Cultura Organizacional, seus processos: Gerência de Projeto e Gerência de Requisitos, suas características: Atributos de Processos AP1.1 e AP2.1 e seu objetivo: Gerenciar Parcialmente o Projeto. A Figura 27, expressa o relacionamento dessas características do Nível G.

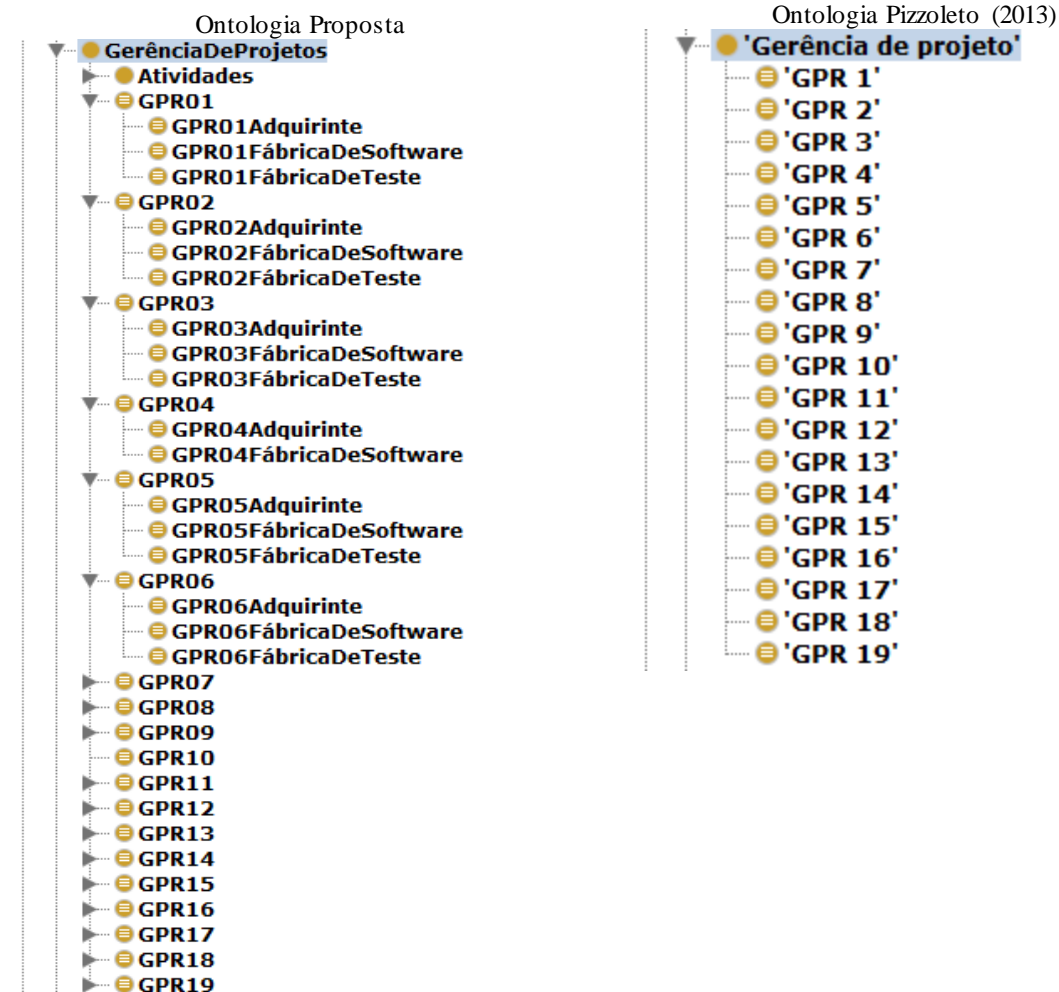
Figura 27: Características do Nível G



5.6 COMPARANDO AS MODELAGENS

Apesar de usar como base a ontologia proposta por Pizzoleto (2013), o trabalho proposto possui algumas diferenças na Modelagem de algumas classes, como por exemplo, na Figura 28.

Figura 28: Comparação da Classe Gerência de Projetos



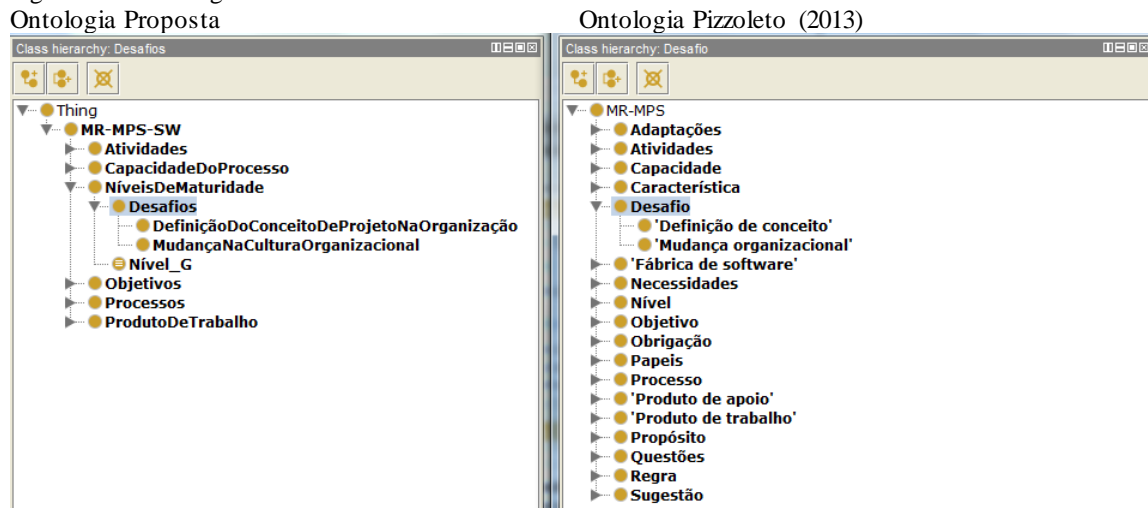
De acordo com a Figura 28, observa-se na modelagem da ontologia proposta, que a Classe Gerência de Projetos, além dos GPRs, possui uma subclasse chamada Atividades. A classe Atividades, foi criada como subclasse da classe Gerência de Projetos, pois de acordo com o Guia Parte 1 de implementação do Nível G, pag. 08, além dos GPRs, a Gerência de Projetos possui algumas atividades como: Desenvolver um plano geral de controle do projeto; Obter o comprometimento e Conhecer o progresso do projeto, que estão descritas nessa classe. Um outro detalhe da modelagem é que cada GPR possui subclasses com resultados direcionados a organizações do tipo: Adquirintes de Software, Fábrica de Teste e Fábrica de Software. Essas subclasses, contém resultados de informações adicionais contidas nos Guias Parte 8, 9 e 10, conforme Tabela 1 direcionados a essas organizações. Outra diferença, está na Figura 28.

Figura 29: Comparando a classe Gerência de Requisitos



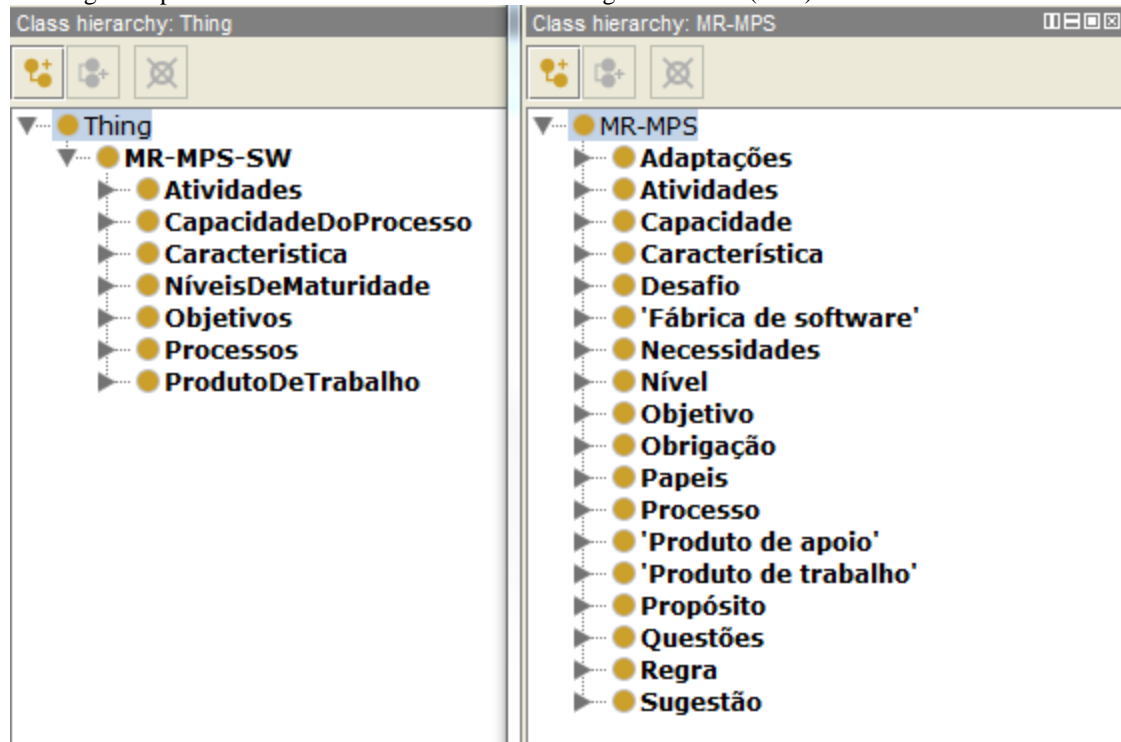
Assim como na classe Gerência de Projetos, observa-se através da Figura 29, que a classe Gerência de Requisitos também teve adicionada aos seus processos de Gerência de Requisitos, subclasses que contém informações adicionais direcionadas a organizações do tipo: Adquirintes de Software, Fábrica de Teste e Fábrica de Software, pois de acordo com os Guias Parte 1, 8, 9 e 10, conforme Tabela 1, essas informações adicionais estão presentes na Gerência de Projetos e Gerência de Requisitos.

Figura 30: Modelagem da Classe Desafios



Conforme Figura 30, na ontologia proposta, a classe Desafios, que contém os desafios presentes no Nível G, foi modelada como sendo uma subclasse da classe Níveis de Maturidade, por entender que os desafios, são subconjuntos presentes nos níveis de maturidade. Na ontologia proposta por Pizzoleto (2013), Desafio é modelado como uma superclasse, quem contém os desafios presentes no Nível G.

Figura 31: Comparação das Classes Principais das Ontologia
 Ontologia Proposta Ontologia Pizzoleto (2013)



A Figura 31, faz uma comparação entre as classes principais presentes na ontologia proposta e na ontologia proposta por Pizzoleto (2013). Observa-se que a ontologia proposta por Pizzoleto (2013) possui um número bem maior de classes que a ontologia proposta. Isso se deve, pelo fato de Pizzoleto (2013) abordar em sua ontologia os níveis de maturidade G e F e contempla-la com inserções de conceitos e terminologias do Guia de Conhecimentos em Gerenciamento de Projetos ou PMBOK (Project Management Body of Knowledge), e do modelo BSC (Balanced Scorecard), um modelo de mensuração de desempenho empresarial para complementar o grupo de indicadores do modelo MPS-SW. Além disso, a ontologia de Pizzoleto (2013) contém sugestões relacionadas aos processos referentes aos Níveis G e F, possui uma classe que define algumas regras a serem seguidas durante a implementação dos processos, aborda algumas questões a serem levantadas durante a implementação dos níveis G e F, dentre outros. Diferente da ontologia proposta por Pizzoleto (2013), a ontologia proposta considerou apenas as estruturas de sete (07) guias, vide Tabela 1, respectivamente, Parte 1, 8, 9, 10, 11, 12 e 13, e levou em consideração, referências contidas em artigos e publicações que abordam o MPS-SW e teve como objetivo, representar apenas os conceitos relevantes às áreas de processos contidas no nível G, de forma a organizar, simplificar, apoiar e facilitar a compreensão dos processos envolvidos nesse nível do guia Melhoria de Processo do Software Brasileiro (MPS-SW).

Apesar de já existir um trabalho relacionado bem próximo ao trabalho proposto, o mesmo optou por construir uma nova ontologia seguindo recomendações do guia prático de construção de ontologias OWL, Horridge (2008). Fez análises, nos trabalhos relacionados, e após considerar algumas modelagens inconsistentes, aplicou essas correções no trabalho proposto e contemplou na ontologia com informações adicionais nos guias direcionado a organizações do tipo Adquirentes de Software, Fábrica de Software e Fábrica de Teste, apresentando uma ontologia alternativa a proposta por Pizzoleto (2013).

6 CONSIDERAÇÕES FINAIS

Apesar do nível G se tratar do estágio inicial para um empresa que busca uma certificação MPS.Br. Trata-se de um nível que possui um grande número de dependências entre seus processos e grande diversidade na quantidade de conteúdo. A forma textual contida nos guias que contemplam esse nível, possuem um grande conjunto de informações que se abrangem a diversos tópicos (processos, atributos, requisitos, informações específicas de acordo com o tipo de organização, etc). Devido a isso, há uma dificuldade de uniformizar o entendimento desses processos por pessoas envolvidas. Neste sentido, o trabalho proposto, buscou organizar, simplificar, apoiar e facilitar a compreensão dos processos envolvidos no nível G do guia Melhoria de Processo do Software Brasileiro (MPS-SW), através de uma representação alternativa desse guia usando ontologia. Apresentou também, uma ontologia para o nível G, alternativa a proposta Pizolleto (2013), corrigiu erros de modelagem em OWL conforme apresentado na seção 5.4 e seguiu a nomenclatura recomendada do guia prático de construção de ontologias OWL, Horridge (2008) para OWL.

Para trabalhos futuros, pretende-se tê-la avaliada por um grupo de especialistas na área, afim de efetuar possíveis correções; Utilizar a representação da ontologia proposta na implantação do nível G em uma empresa; procurar mecanismos automáticos para a implantação do nível G por agentes, podendo utilizar o Raciocinador e Estender a modelagem para o nível F;

Espera-se que o uso da ontologia seja bastante útil, servindo de apoio não só para organizações desenvolvedoras de software que buscam implementar esse nível, mais também, para aqueles profissionais envolvidos nesse processo de implementação, pois através da ontologia proposta, é possível verificar quais são os processos envolvidos no Nível G, quais os resultados esperados de cada processo, quais as dependências entre esses processos, como

eles se relacionam, dentre outros. Neste sentido, a criação de uma ontologia aplicada a esse nível, torna-se uma ferramenta de apoio que busca ajudar as pessoas a compreenderem melhor os passos a serem seguidos durante a implementação do nível G.

-

-

7 REFERÊNCIAS

ALMEIDA, M. B.; BAX, M. P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. **Ciência da Informação**, Brasília, DF, v. 32, n. 3, p. 7-20, set./dez. 2003.

BRANDÃO LIMA, P.S. **Proposta de Um Modelo Simplificado de Aquisição de Software para Pequenas Empresas**. 2004. Dissertação de Mestrado - Escola Politécnica, Universidade de São Paulo, 2004.

BREITMAN, Karin Koogan. *Web semântica: a internet do futuro*. Rio de Janeiro: LTC, 2006.

BORST, W. N. *Construction of Engineering Ontologies*. University of Twente – Centre for Telematica and Information Technology, Enschede, Nederland, 1997.

CUNHA, I. B. A. Dias K. J. A, Resende, J. H. C. Dificuldades Encontradas Na Implementação Mps.Br nível G: Estudo de Caso. Artigo publicado em e-xacta, Belo Horizonte, v.4, n. 3, p. 19-35. (2011). Editora UniBH. Disponível em: www.unibh.br/revistas/exacta/ Acesso em: 26 out. 2013.

DAL MORO R., Falbo R. A. Uma Ontologia para o Domínio de Qualidade de Software com Foco em Produtos e Processos de Software. *third Workshop on Ontologies and Metamodeling in Software and Data Engineering - WOMSDE 2008*

FALBO, R. A. Experiences in using a method for building domain ontologies. In: *International Workshop on Ontology in Action. Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering*, pages 474–477. 2004.

GRUBER, T. What is an ontology <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>, acesso em 10out/2013.

GRUBER, T. A translation approach to portable Ontologies. *Knowledge Acquisition, California*, v. 5, n. 2, p. 199-220, Apr. 1993.

GUARINO, N. Formal ontology and information systems. In *Proceedings of International Conference in Formal Ontology and Information Systems - FOIS'98*, pages 3–15. 1998.

CHANDRASEKARAN, B.; JOSEPHSON, J. R.; BENJAMINS, V. R. What are ontologies, and why do we need them? *Intelligent Systems and their Applications*, IEEE, Columbus, v. 14, n. 1, p. 20-26, Jan./Feb. 1999

HORRIDGE, M. et al. Um guia prático para a construção de ontologias OWL, plugin Protégé_OWL 3.4. Trad. SOARES, D.R; ALMEIDA, M. B. Disponível em: < www.eci.ufmg.br/mba/>, 2008. Acesso em: 12 fev 2014.

KIRYAKOV, A. Ontologies for Knowledge Management. In: DAVIES, J.; et al. (eds). *Semantic Web Technologies: trends and research in ontology-based systems*, p. 115-138, 2006

KOSCIANK, 2007, Andre; SOARES, Michel S. Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2ª Edição. São Paulo: Novatec Editora, 2007.

MPS.BR Melhoria de processo do software brasileiro: Website do programa MPS, 2012.

Disponível em: <<http://www.softex.br/mpsbr>>. Acesso em: 10 abr. 2014.

NOY, Natalya F.; McGUINNESS, Deborah L. Ontology Development 101: a guide to Creating your first ontology. Disponível em:

<<http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf>>.

Acesso em: 13 fev. de 2014.

Ontology development 101: a guide to creating your first ontology. 2006. Disponível em: <http://protege.stanford.edu/publications/ontology_development/ontology101noymcguinness.html>.

Acesso em: 14 nov. 2013.

Ontology development 101: a guide to creating your first ontology. 2006.

Disponível em: http://protege.stanford.edu/publications/ontology_development/ontology101-noymcguinness.html>. Acesso em: 14 nov. 2013.

PIZZOLETO, A. V., OLIVEIRA, H. C., OLIVEIRA, C. S. An ontology on the level G of the Software Process Model MPS.Br to assist business processes modeling. In: International Conference WWW/Internet, Madrid. 2012.

Pizzoleto, Alessandro Viola. Ontologia empresarial do modelo de referência MPS para software (MR-MPS-SW) com foco nos níveis G e F / Alessandro Viola Pizzoleto. - São José do Rio Preto: [s.n.], 2013. 147 f. : il. ; 30 cm.

SOFTEX, 2009 ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. MPS.BR – Guia Geral. 2009 Disponível em: <<http://www.softex.com.br>> Acesso em: 16 abr. 2014.

SOFTEX. MPS.BR – Melhoria de Processo do Software Brasileiro, Guia de Implementação – Parte 1: Nível G (Versão 1.1). Disponível em <<http://www.softex.br>> Acesso em: 04 abr 2014.

SOFTEX. MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral (v1.2), SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2009. Disponível em <<http://www.softex.br>> Acesso em: 04 abr 2013.

SOFTEX, 2012 ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. MPS.BR – Guia Geral MPS de Software: 2012. Disponível em: www.softex.br Acesso em: 04 abr 2013.

SOFTEX - Associação para Promoção da Excelência do Software Brasileiro. MPS.BR Melhoria de processo do software brasileiro: guia de implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS. 2011. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_2_2011.pdf>. Acesso em: 15 abr. 2013.

SANTAREM SEGUNDO, J E. Representação Iterativa: um modelo para repositórios digitais. 2010. 224f. Tese (Doutorado em Ciência da Informação) – Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, Marília, 2010.

SOUZA R. F., FALBO R. A. NANDAMUDI L. V. XII Workshop de Computação Aplicada - WORCAP 2012.